CISPA
HELMHOLTZ CENTER FOR INFORMATION SECURITY

Consensus in the presence of Overlapping Faults and Total Omission Setting

Julian Loss, CISPA Helmholtz Center for Information Security

Kecheng Shi, CISPA Helmholtz Center for Information Security

Gilad Stern, Tel Aviv University

Funded by
the European Union

Table of Contents

Byzantine Agreement

Byzantine Agreement

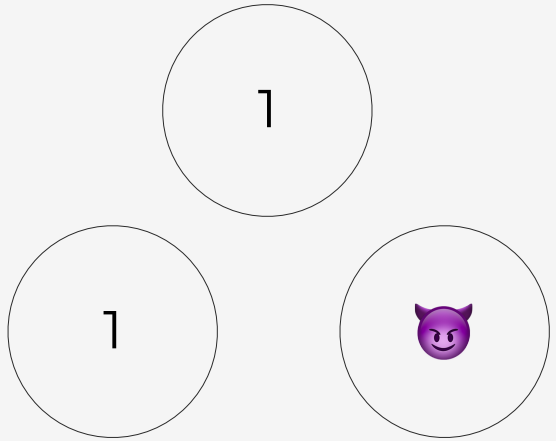
Inputs

- Validity
- Consistency
- Termination

Outputs

Byzantine Agreement

Inputs

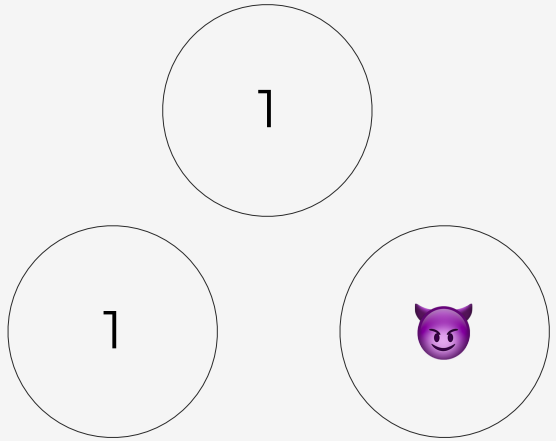


- Validity
- Consistency
- Termination

Outputs

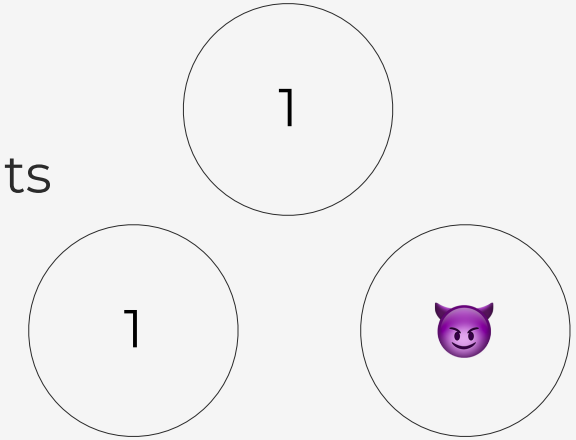
Byzantine Agreement

Inputs



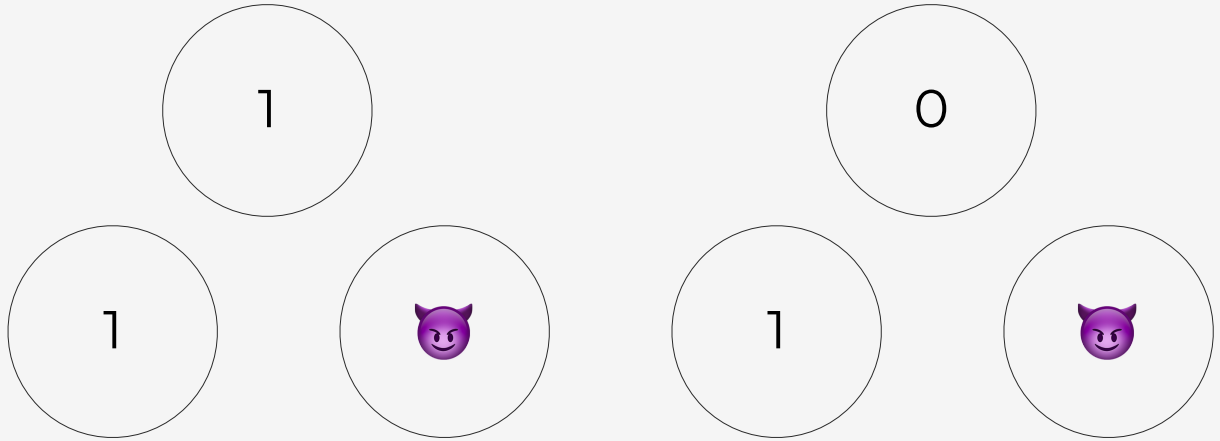
- Validity
- Consistency
- Termination

Outputs



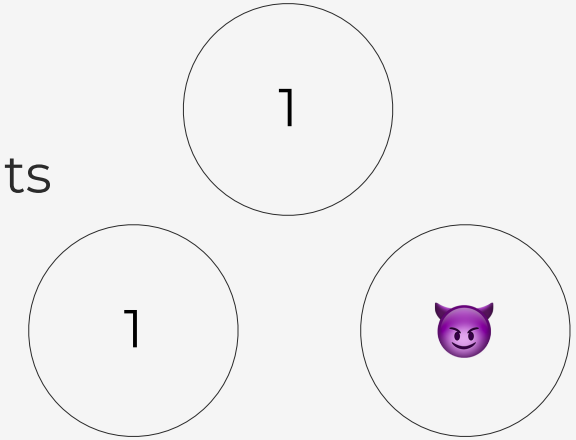
Byzantine Agreement

Inputs



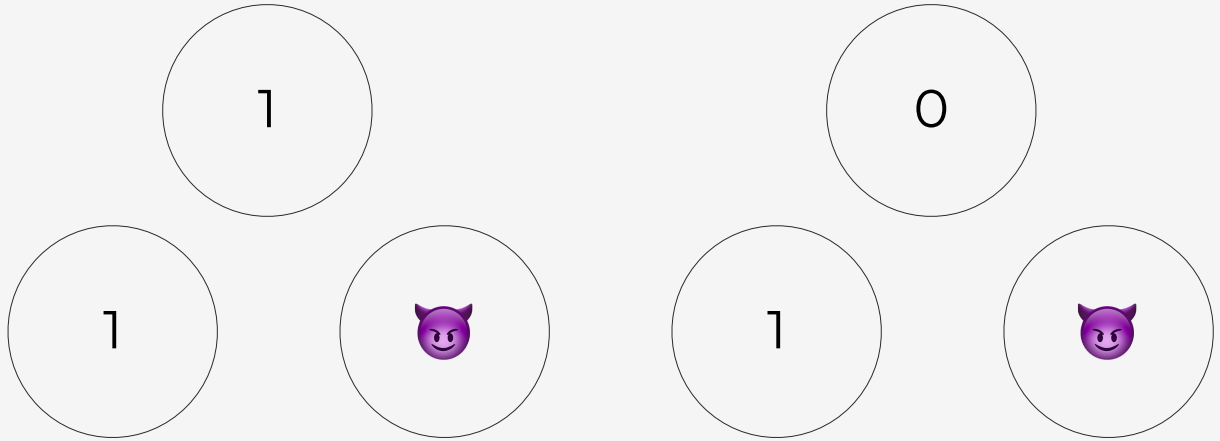
- Validity
- Consistency
- Termination

Outputs



Byzantine Agreement

Inputs

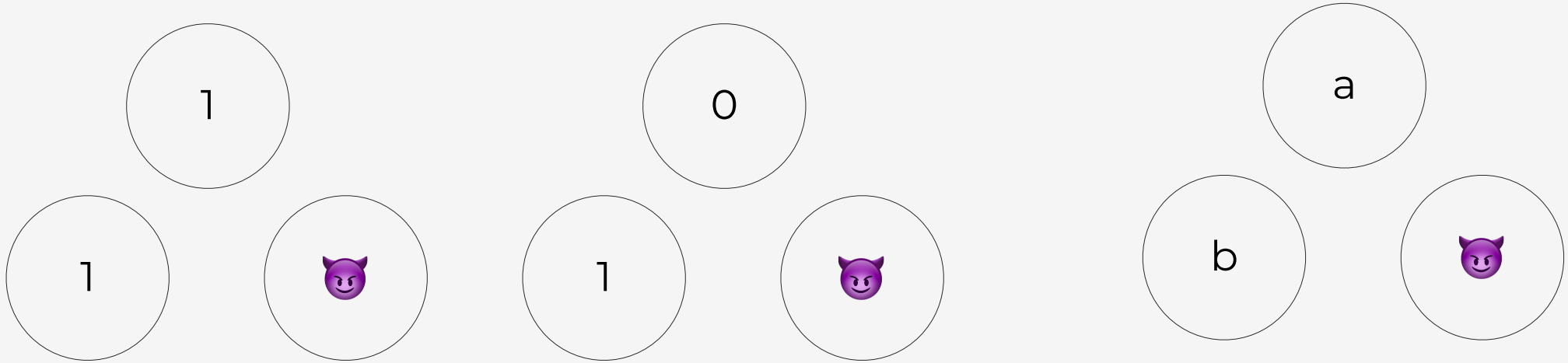


- Validity
- Consistency
- Termination

Outputs

Byzantine Agreement

Inputs

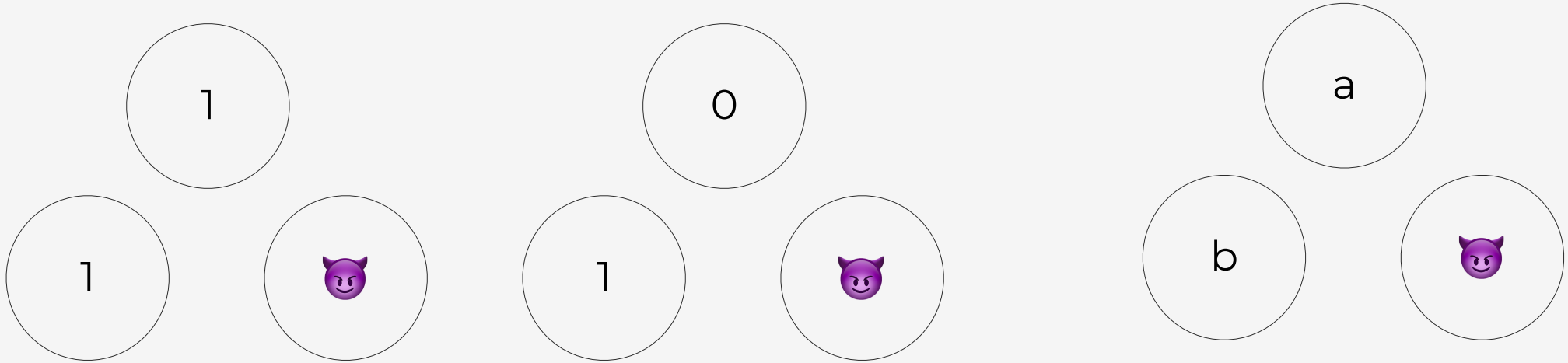


- Validity
- Consistency
- Termination

Outputs

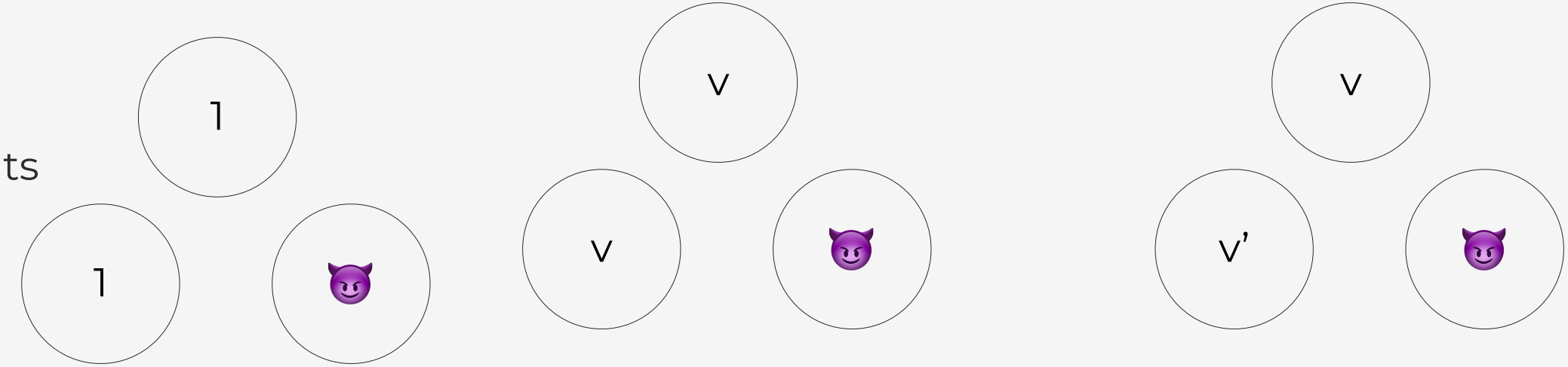
Byzantine Agreement

Inputs



- Validity
- Consistency
- Termination

Outputs



Mixed Fault Model

Mixed Fault Model

- Send Omission(Ghost👻)
- Receive Omission(Zombie🧟)
- Byzantine Corruption
- Full-Omission(Overlapping Faults)

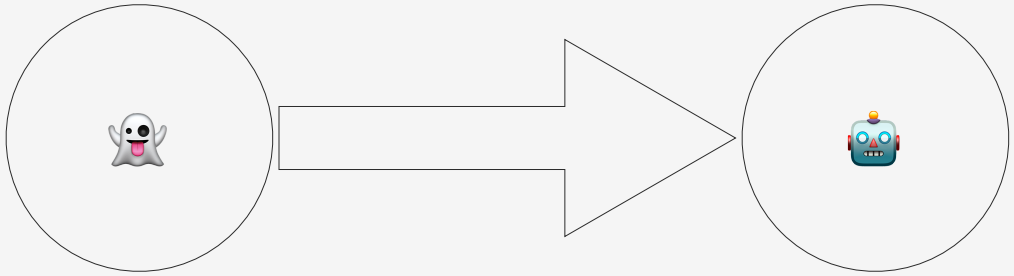
Omissions

Omissions

- Send Omission 👻

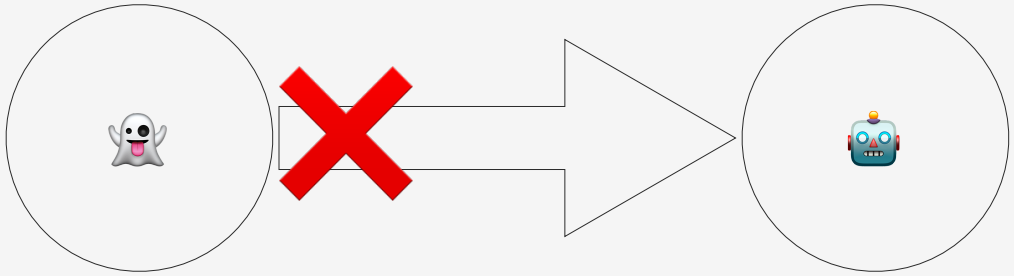
Omissions

- Send Omission 👻



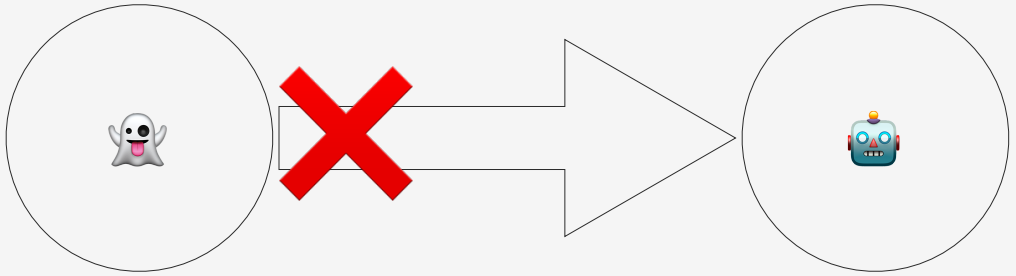
Omissions

- Send Omission 👻

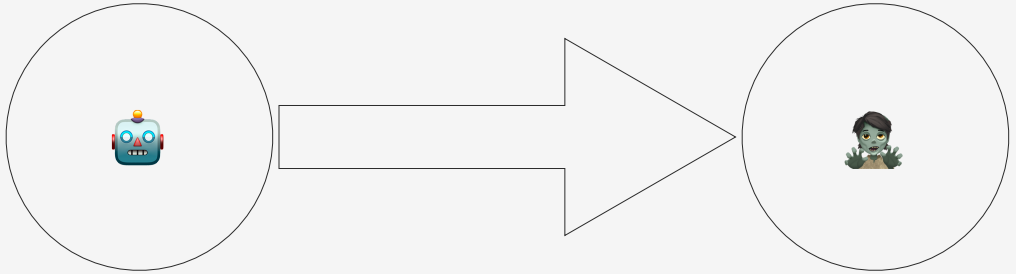


Omissions

- Send Omission 👻

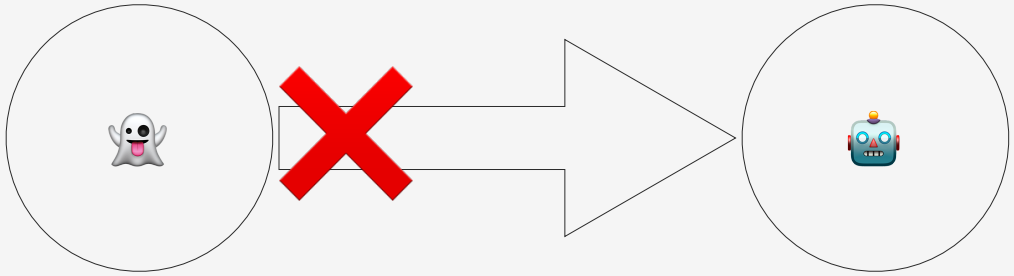


- Receive Omission 🧟

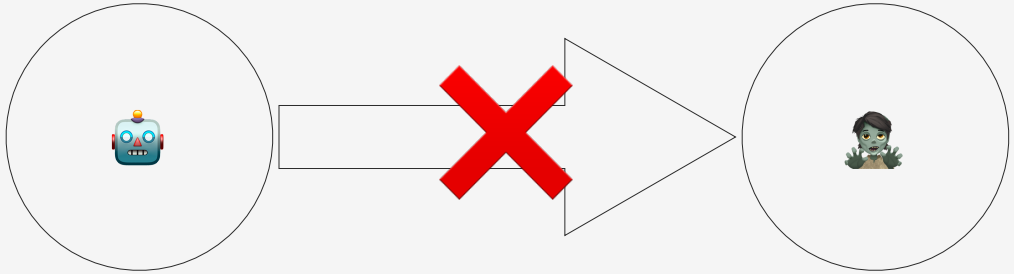


Omissions

- Send Omission 👻



- Receive Omission 🧟



Variant of Uniformity (Undead)

- Receive Omission

Variant of Uniformity (Undead)

- Receive Omission

  It is allowed to output a flag Z=True to indicate it is actually receive omission.

  When it outputs Z=True, it must be receive omission.

  When it does not detect itself as receive omission, it has to output the “correct” value.

Variant of Uniformity (Undead)

- Receive Omission

  It is allowed to output a flag Z=True to indicate it is actually receive omission.

  When it outputs Z=True, it must be receive omission.

  When it does not detect itself as receive omission, it has to output the “correct” value.

- Send Omission

Variant of Uniformity (Undead)

- Receive Omission

  It is allowed to output a flag Z=True to indicate it is actually receive omission.

  When it outputs Z=True, it must be receive omission.

  When it does not detect itself as receive omission, it has to output the “correct” value.

- Send Omission

  When it sets itself as Send Omission, it must be the case it is send omission.

  It must output the “correct” value if it is not receive omission.

Basic Idea: Self-Detection

Basic Idea: Self-Detection

- **Receive Omission Detection**

Basic Idea: Self-Detection

- **Receive Omission Detection**

If a party receives number of messages less than it expected, at least one message has been dropped on its side, so it can detect itself as receive omission.

Basic Idea: Self-Detection

- **Receive Omission Detection**

If a party receives number of messages less than it expected, at least one message has been dropped on its side, so it can detect itself as receive omission.

- **Send Omission Detection**

Basic Idea: Self-Detection

- **Receive Omission Detection**

If a party receives number of messages less than it expected, at least one message has been dropped on its side, so it can detect itself as receive omission.

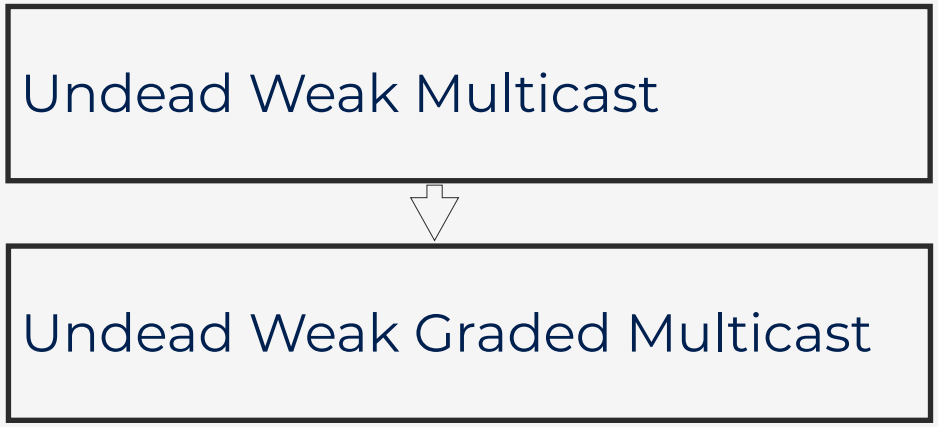
- **Send Omission Detection**

If a party knows at least one nonfaulty party has not received the message it is expected to send, it can detect itself as send omission.

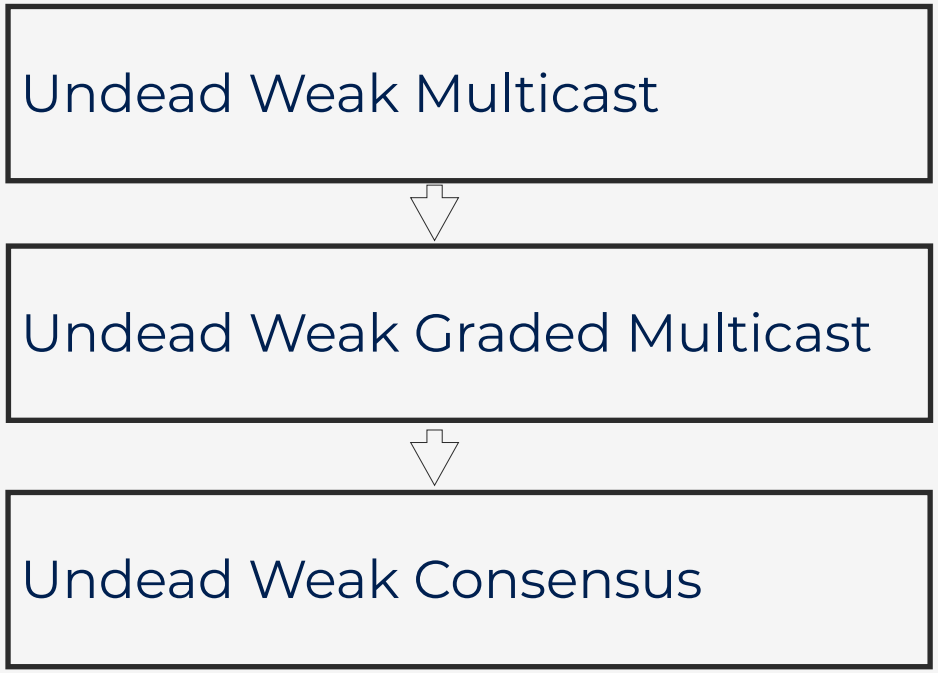
Our Starting Point: Zombies and Ghosts[LS2022]

Undead Weak Multicast

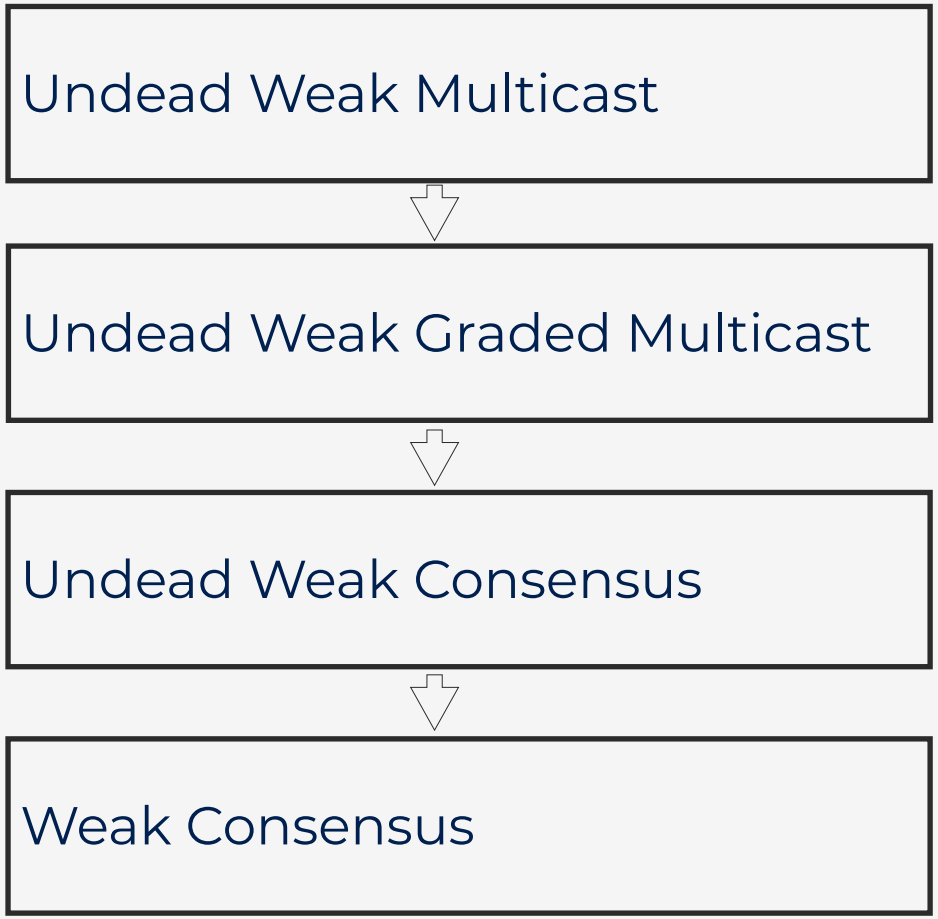
Our Starting Point: Zombies and Ghosts[LS2022]



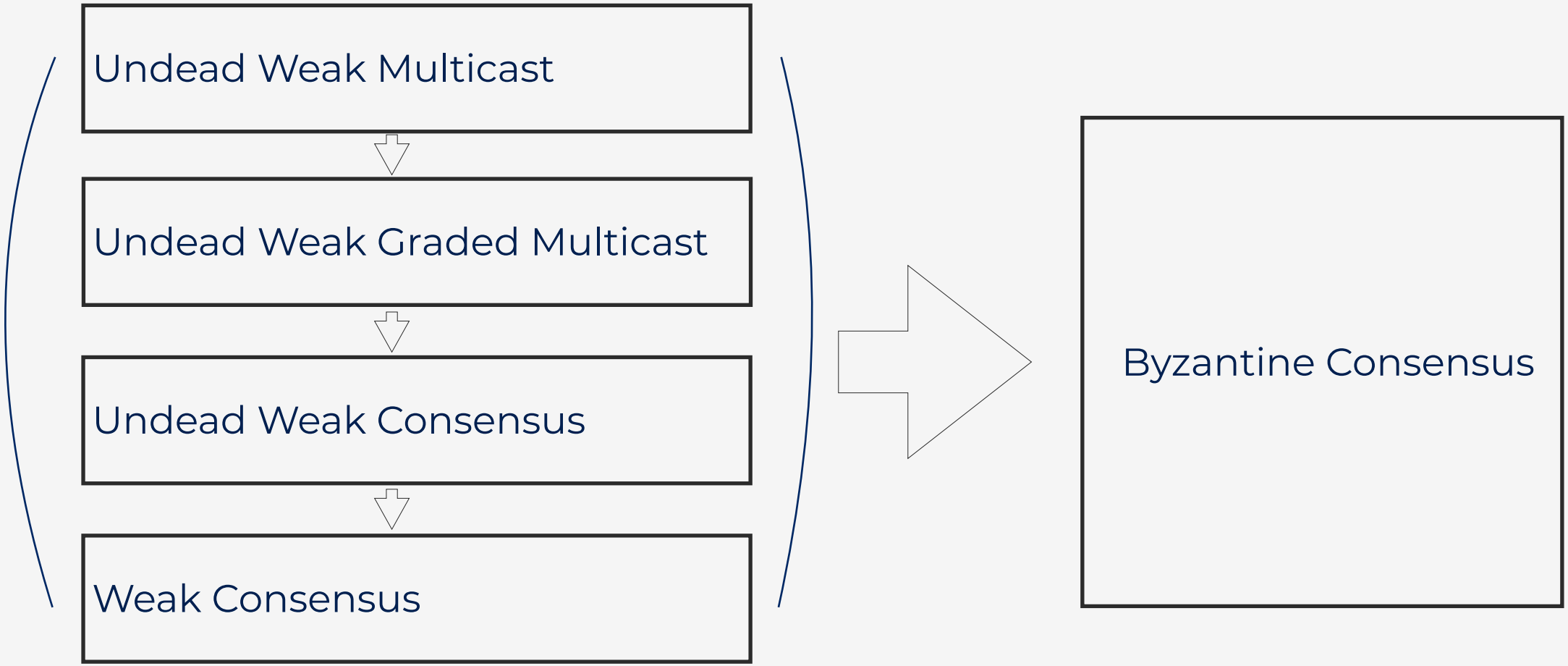
Our Starting Point: Zombies and Ghosts[LS2022]



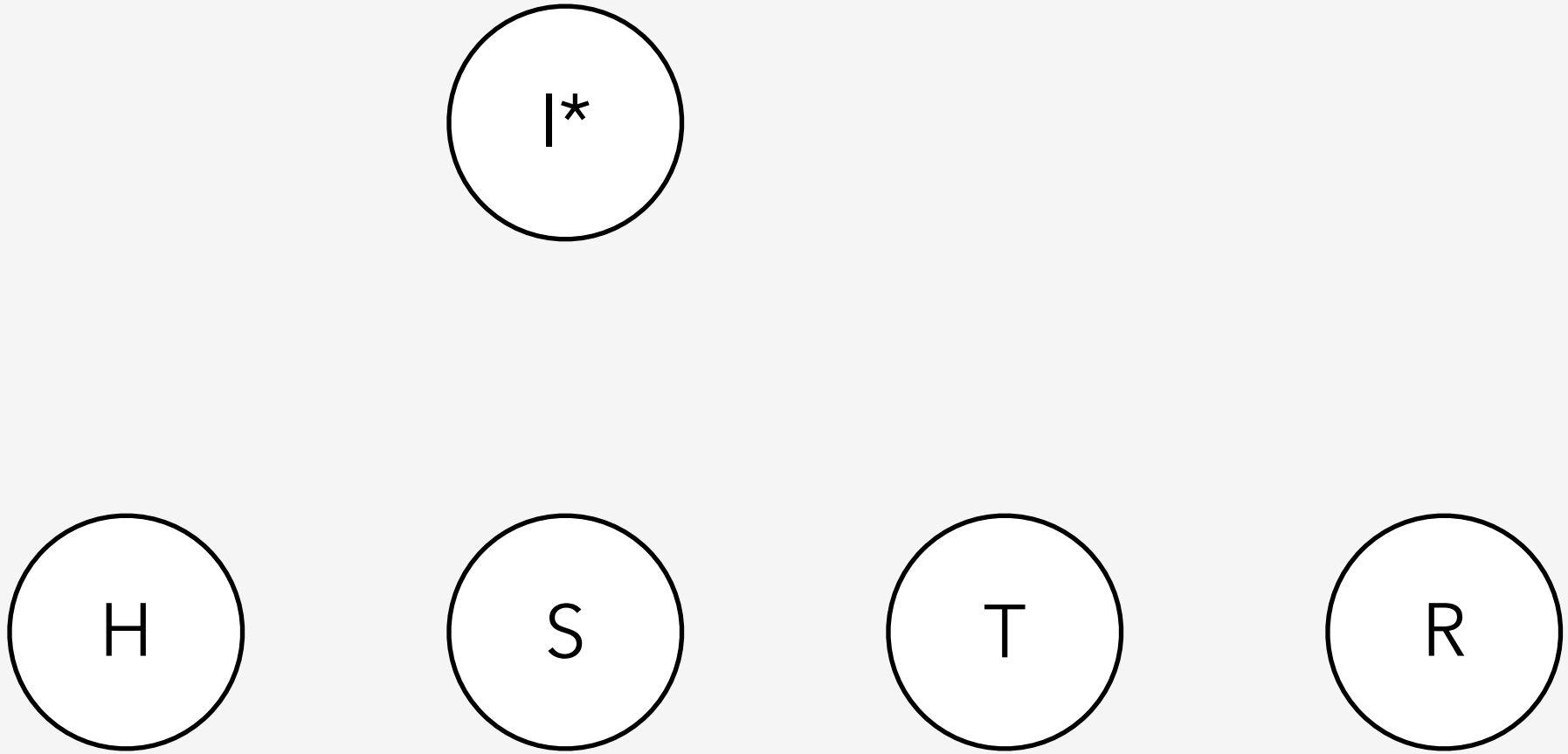
Our Starting Point: Zombies and Ghosts[LS2022]



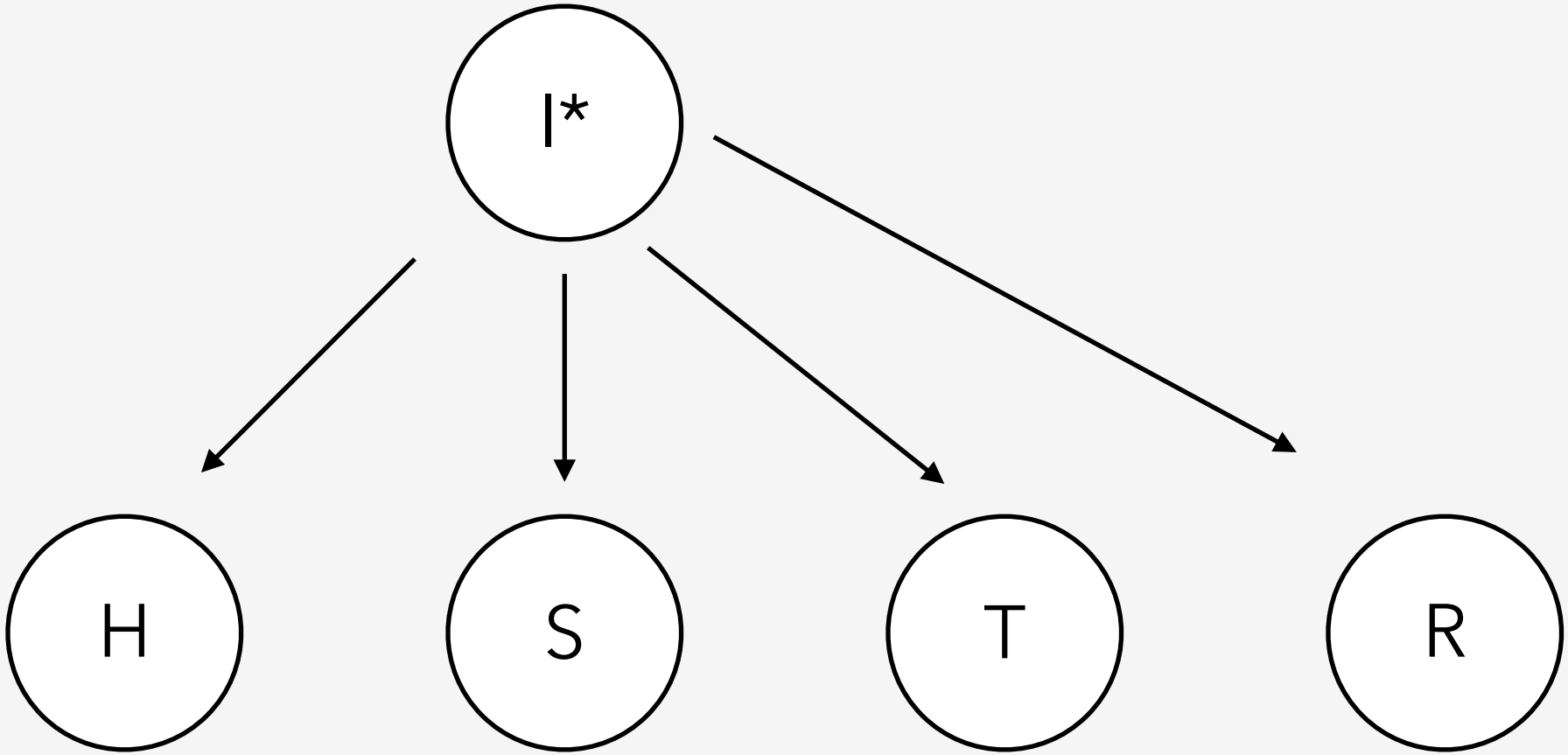
Our Starting Point: Zombies and Ghosts[LS2022]



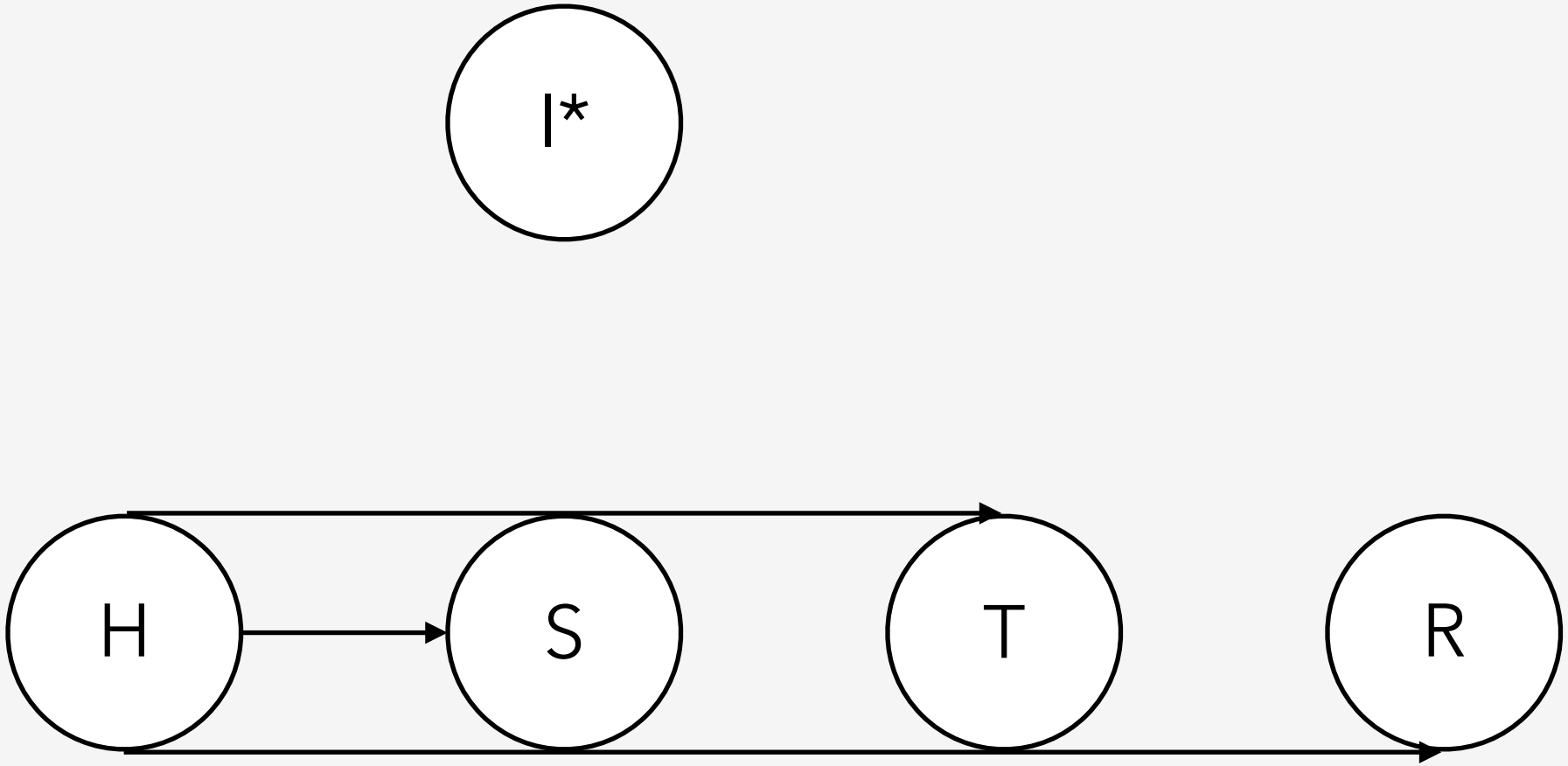
Undead Weak Multicast[LS23]



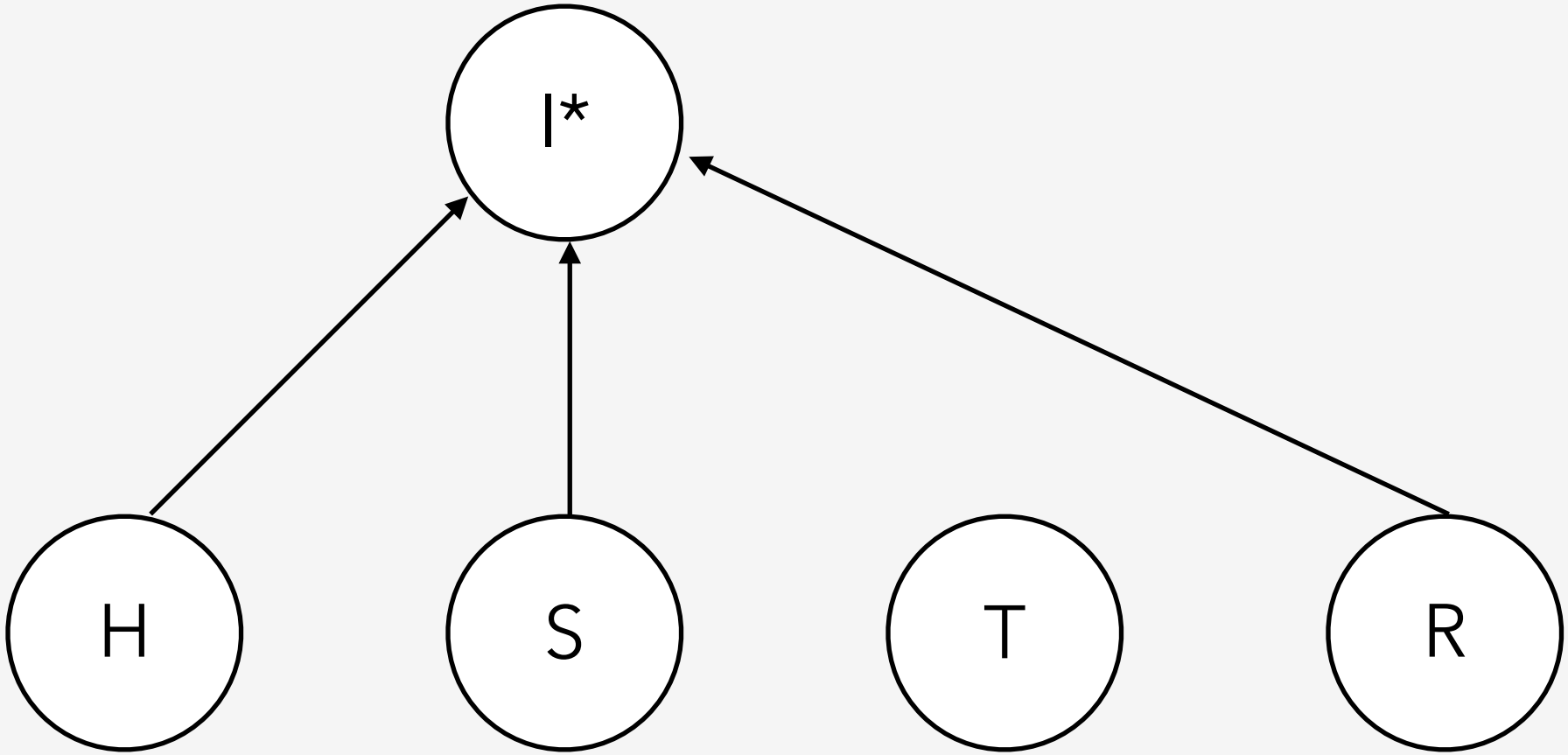
Undead Weak Multicast[LS23]



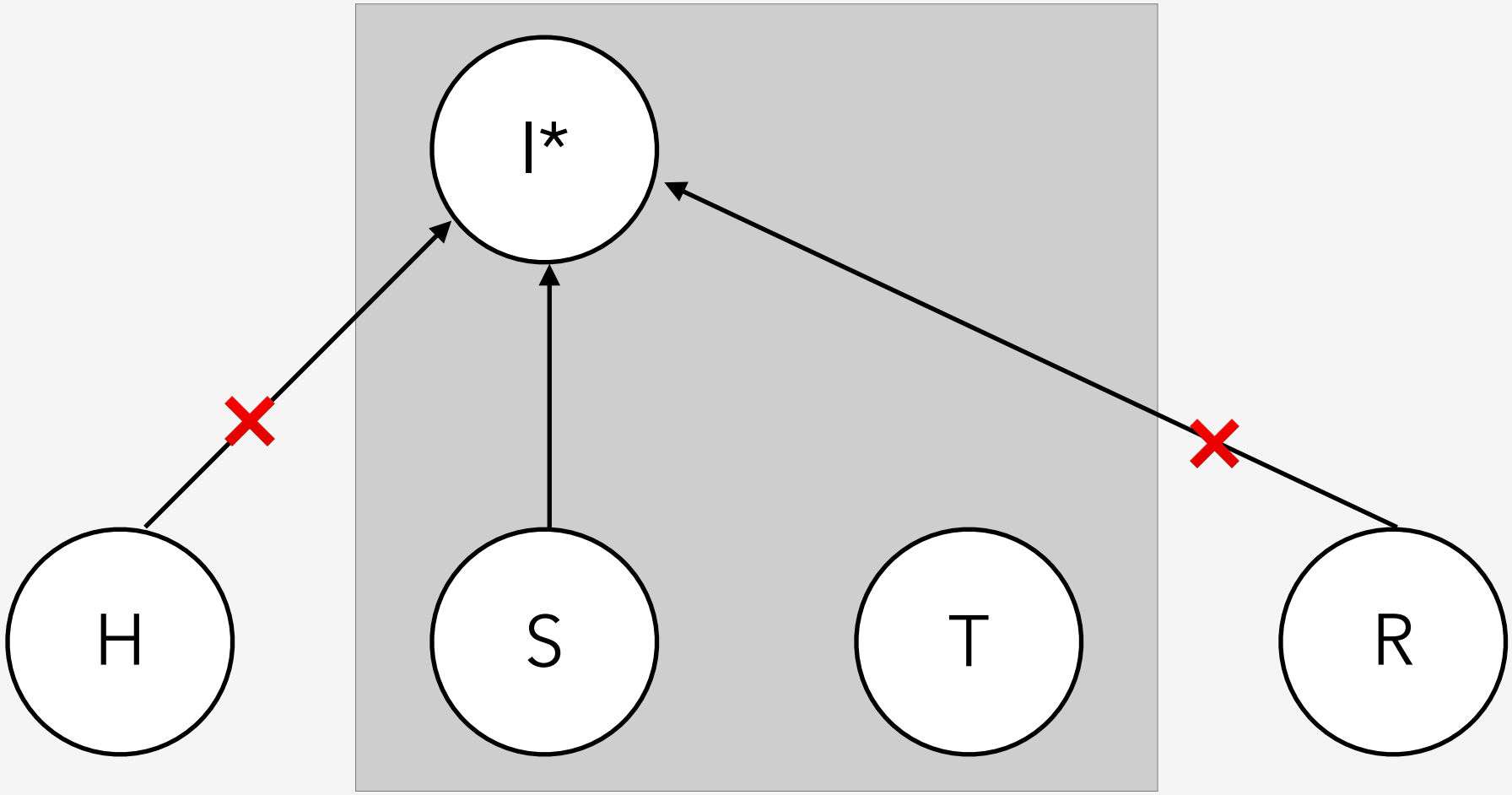
Undead Weak Multicast[LS23]



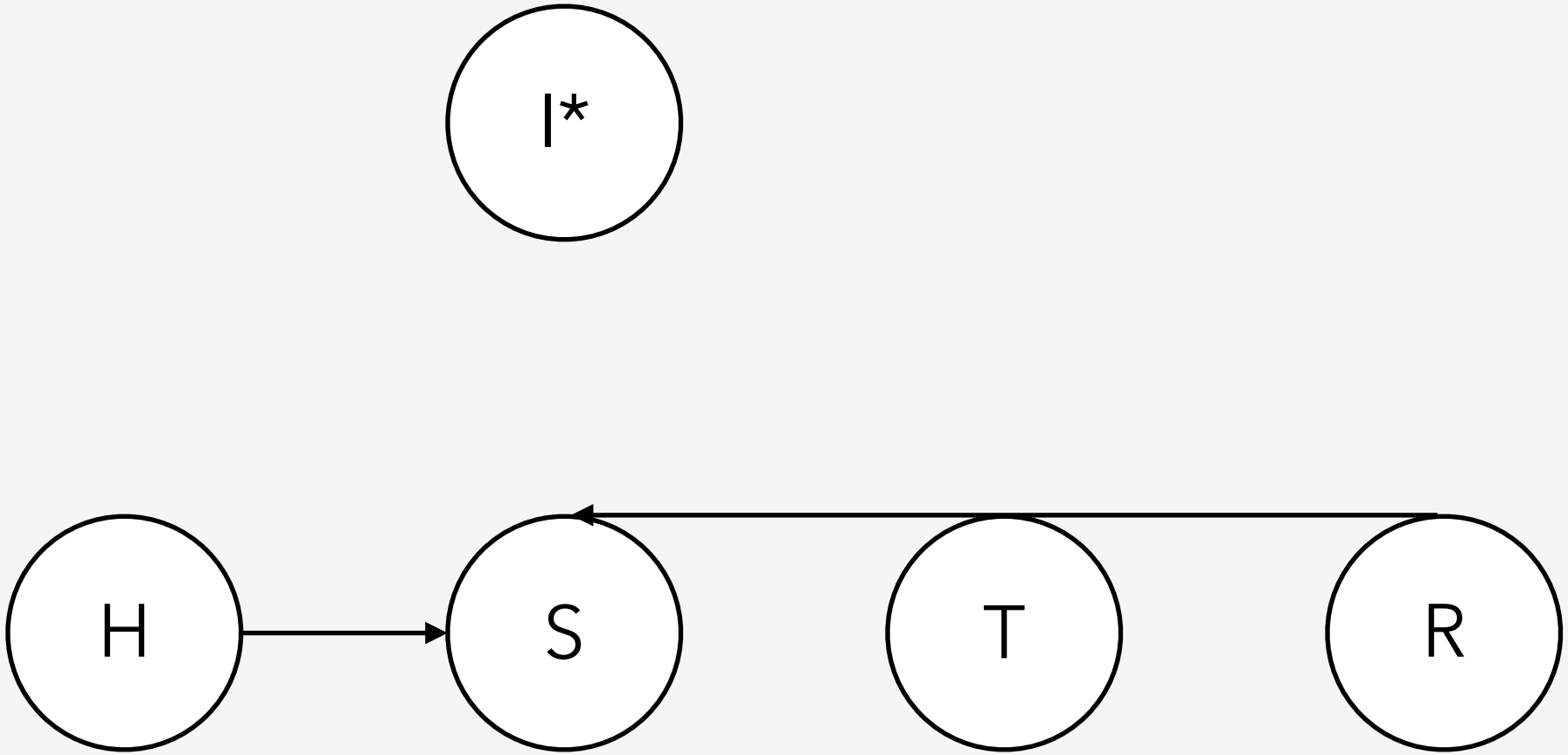
Undead Weak Multicast[LS23]



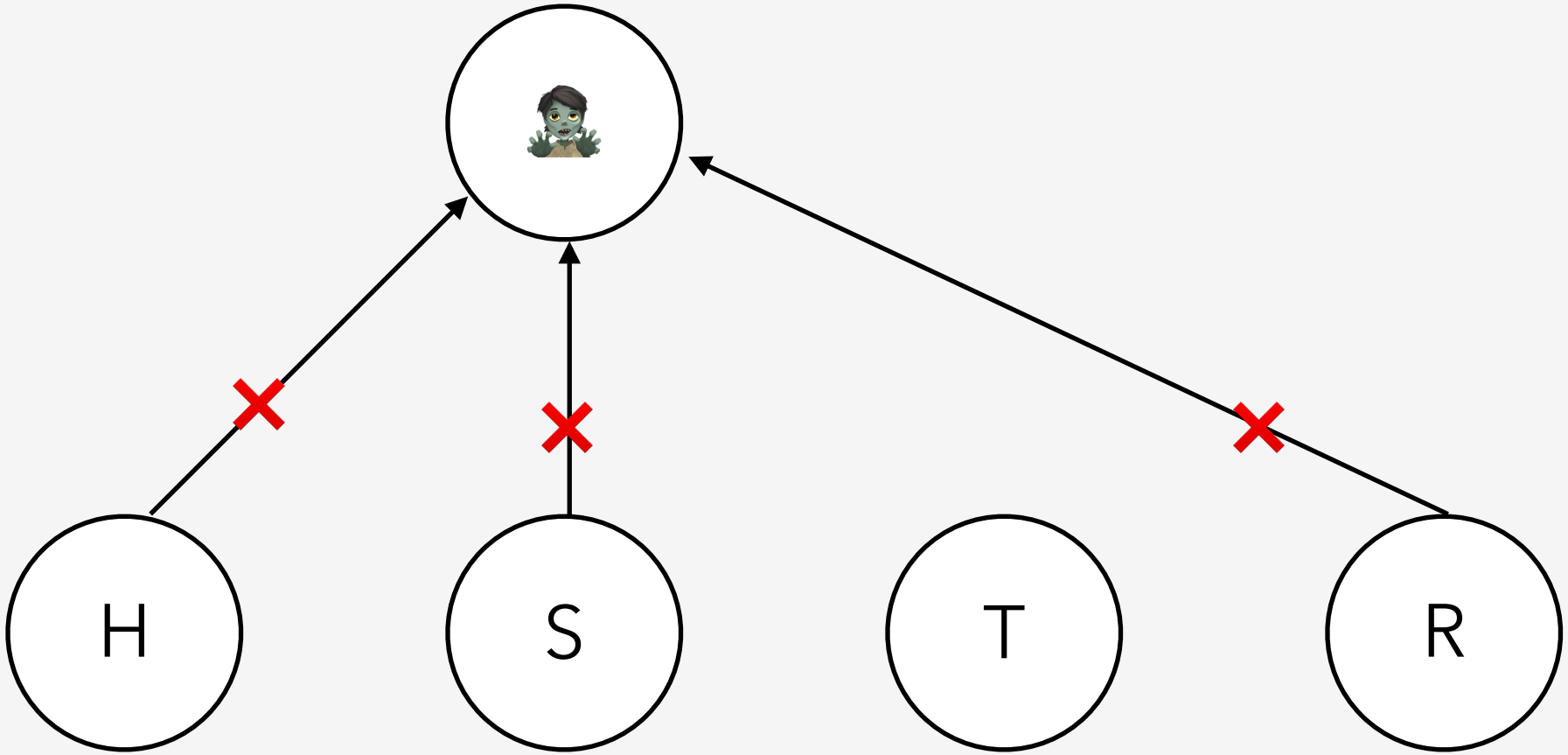
Failed to detect a Full-Omission Sender



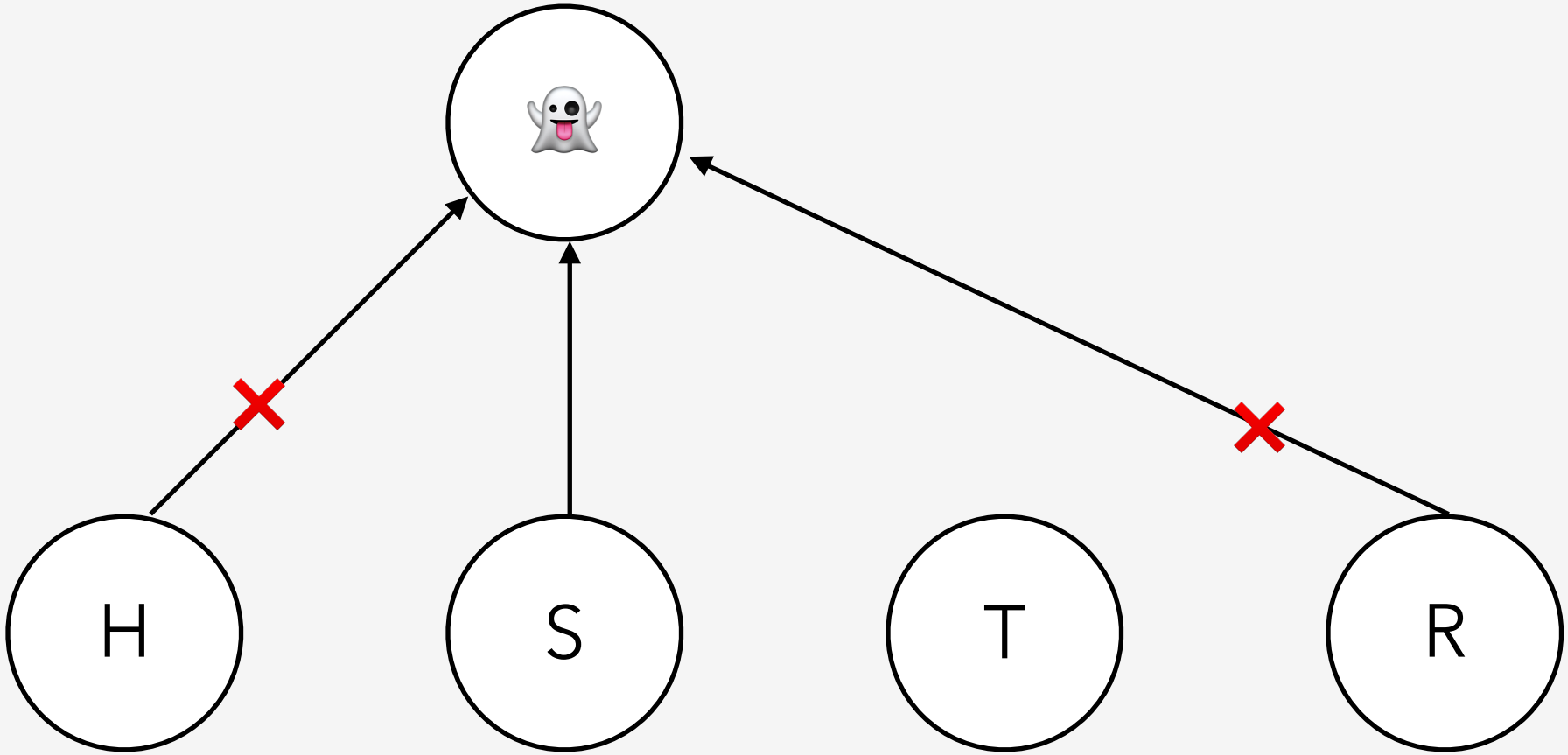
Our Solution Against Overlapping Faults



Our Solution Against Overlapping Faults



Our Solution Against Overlapping Faults



Second Starting Point: Previous Lower Bound [ELT22]

Second Starting Point: Previous Lower Bound [ELT22]

If receive omissions are forced to become zombies and stop participating from the beginning of the protocol, there is no Byzantine Agreement Protocol with $2t + s + r \geq n$.

Second Starting Point: Previous Lower Bound [ELT22]

If receive omissions are forced to become zombies and stop participating from the beginning of the protocol, there is no Byzantine Agreement Protocol with $2t + s + r \geq n.$

Closes the Door, Leaves a Window

Second Starting Point: Previous Lower Bound [ELT22]

If receive omissions are forced to become zombies and stop participating from the beginning of the protocol, there is no Byzantine Agreement Protocol with $2t + s + r \geq n$.

Closes the Door, Leaves a Window

Our Result: There is a Byzantine Agreement Protocol with $s + r = n$.

A Consensus Protocol with s+r=n(s<n)

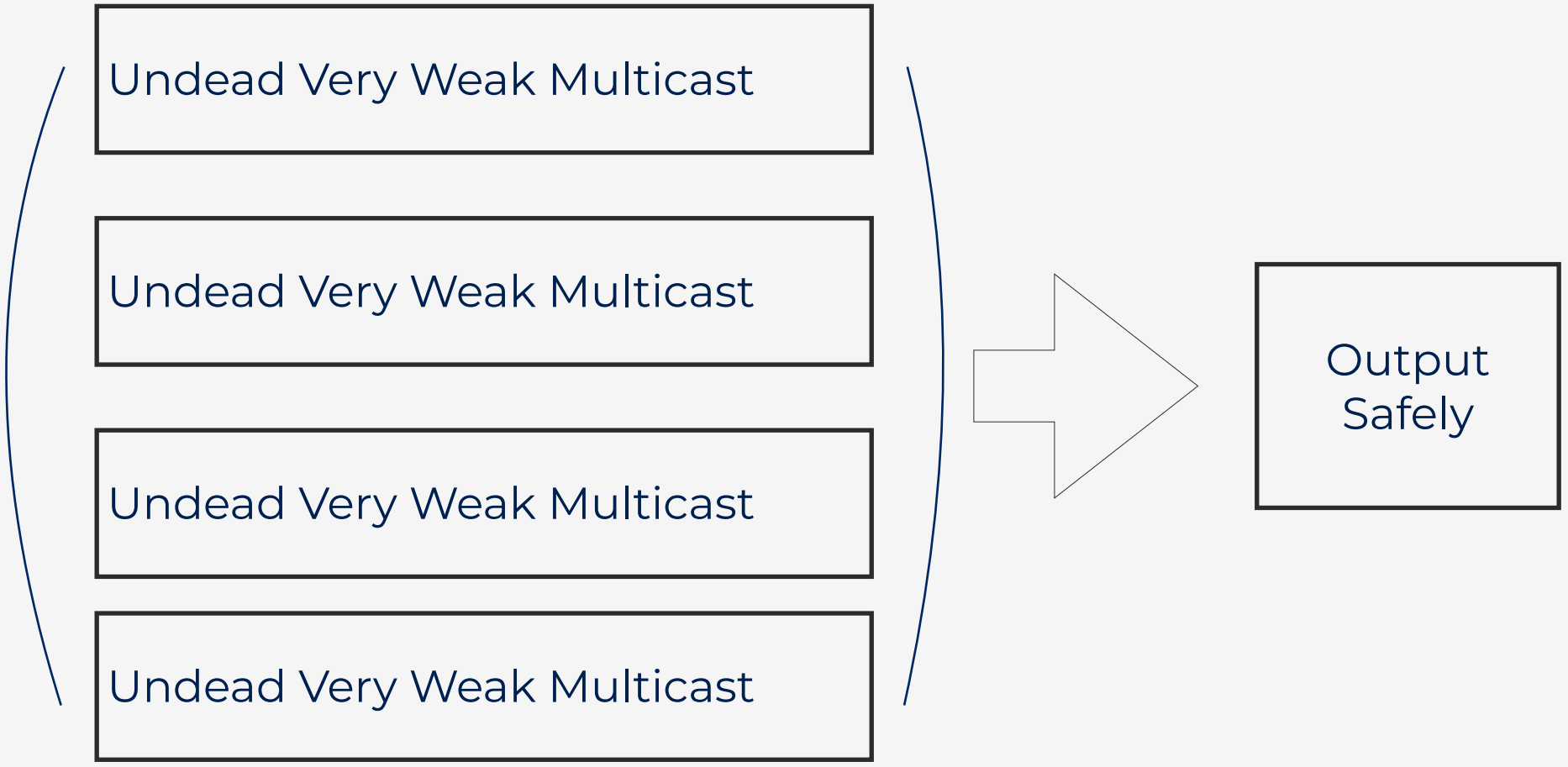
A Consensus Protocol with s+r=n(s<n)

Undead Very Weak Multicast

A Consensus Protocol with s+r=n(s<n)

For s+1 Different Leaders

A Consensus Protocol with s+r=n(s<n)



For s+1 Different Leaders

Total Omission Setting (s+r=n)

Total Omission Setting (s+r=n)

- **Complete Characterization**

Impossibility proof in the case $s = n$.

Total Omission Setting (s+r=n)

- **Complete Characterization**

  Impossibility proof in the case $s = n$.

- **Separation Between Agreement and Broadcast**

  Impossibility proof for Broadcast with $s + r = n$.

Total Omission Setting (s+r=n)

- **Complete Characterization**

  Impossibility proof in the case $s = n$.

- **Separation Between Agreement and Broadcast**

  Impossibility proof for Broadcast with $s + r = n$.

- **Optimal Resilience**

  Impossibility proof for Agreement with $s + r > n$ (with overlapping).

Conclusion

Conclusion

- An Improvement of the state-of-the-art Mixed-Fault Byzantine Agreement Protocol in the presence of Overlapping Faults.

Conclusion

- An Improvement of the state-of-the-art Mixed-Fault Byzantine Agreement Protocol in the presence of Overlapping Faults.
- An Optimal Byzantine Agreement Protocol in the Total Omission Setting.

Conclusion

- An Improvement of the state-of-the-art Mixed-Fault Byzantine Agreement Protocol in the presence of Overlapping Faults.
- An Optimal Byzantine Agreement Protocol in the Total Omission Setting.
- Many Interesting features in the Total Omission Setting.

Conclusion

- An Improvement of the state-of-the-art Mixed-Fault Byzantine Agreement Protocol in the presence of Overlapping Faults.
- An Optimal Byzantine Agreement Protocol in the Total Omission Setting.
- Many Interesting features in the Total Omission Setting.

Thank you!