Lightning Network Analysis and Recursive Virtual Channels

Orfeas Stefanos Thyfronitis Litos
University of Edinburgh
8/2/2021

20,000 tx/s

7 tx/s

Problem

All txs validated by all wallets

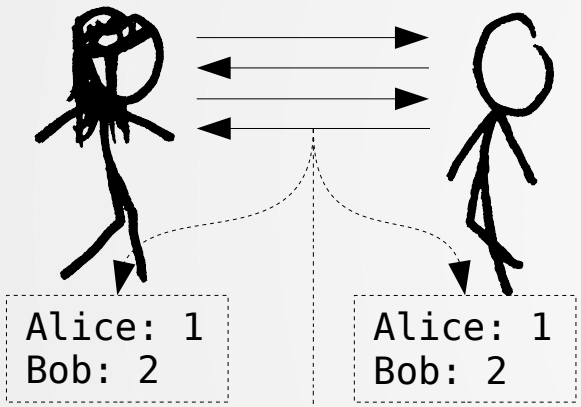
Solution

- Move most txs off-chain
- Resolve disputes on-chain

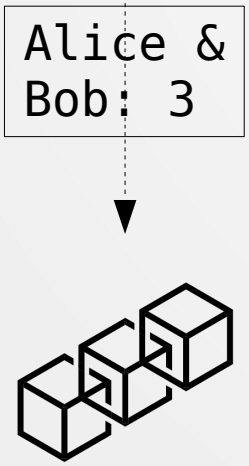
Part 1
A Composable Security Treatment of the Lightning Network
[CSF 2020]

Lightning Channels

Open

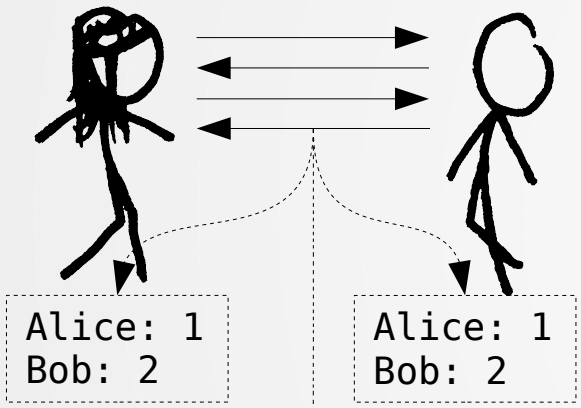


F

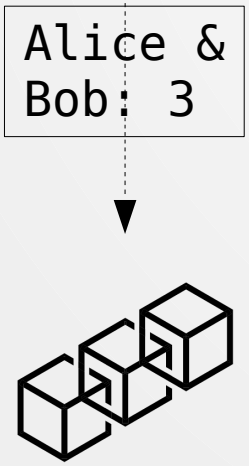


Lightning Channels

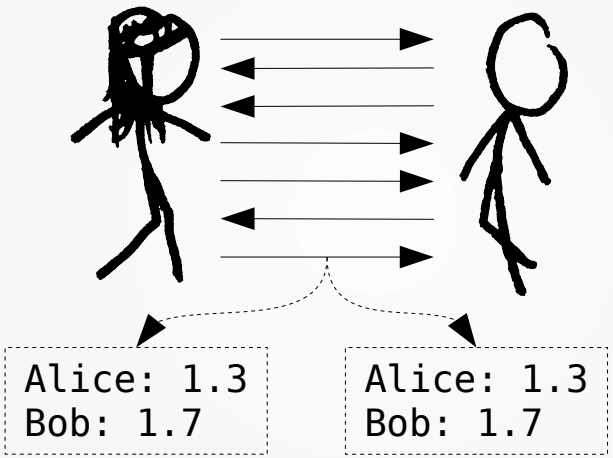
Open

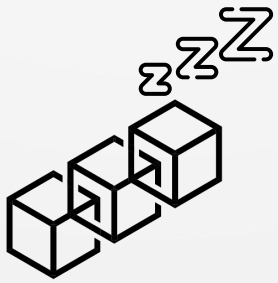


F



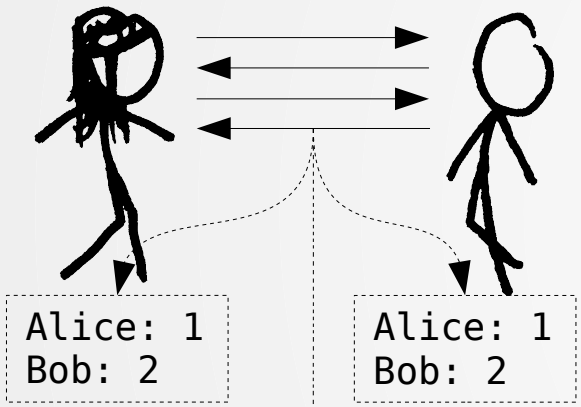
Pay



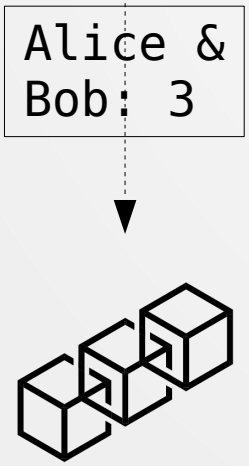


Lightning Channels

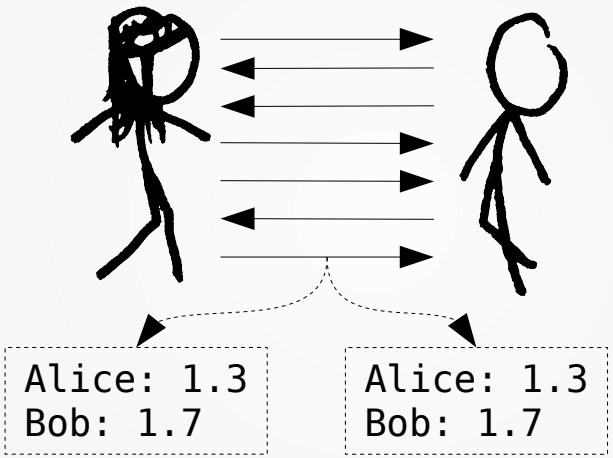
Open



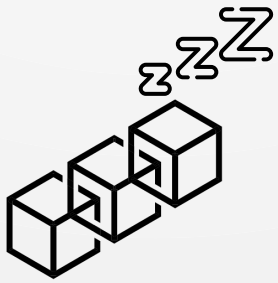
F



Pay

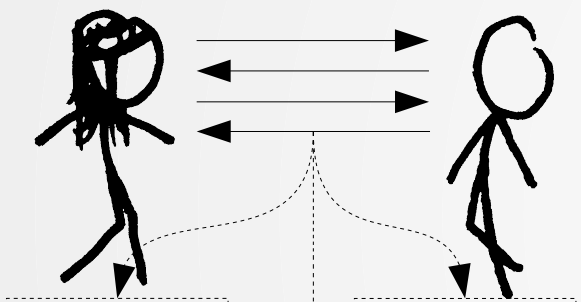


- Unlimited times
- No touching blockchain



Lightning Channels

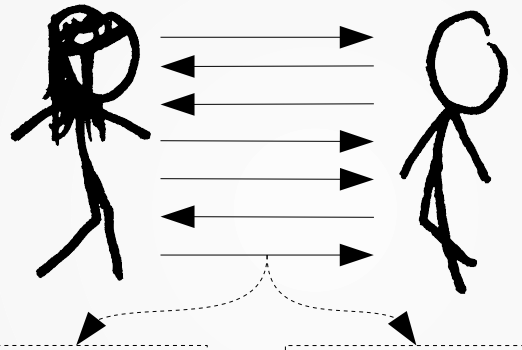
Open



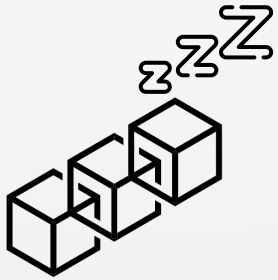
F

Alice &
Bob: 3

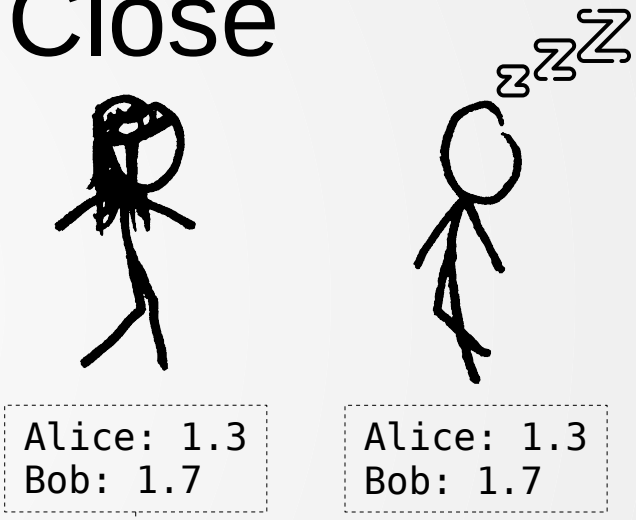
Pay



- Unlimited times
- No touching blockchain

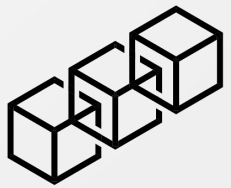


Close



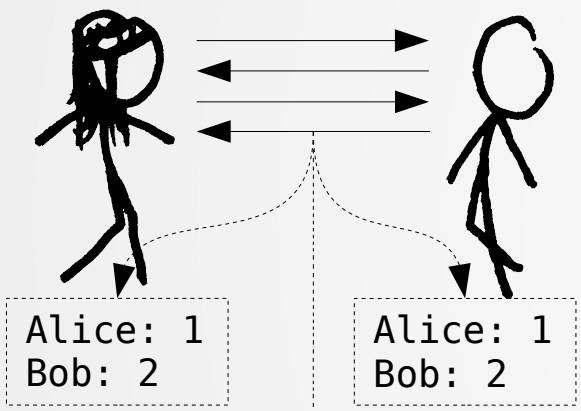
C

Alice: 1.3
Bob: 1.7



Lightning Channels

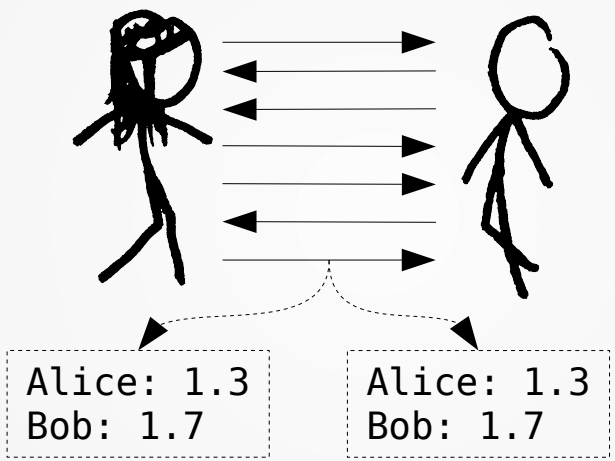
Open



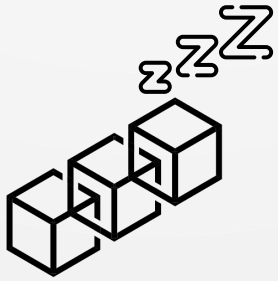
F

Alice &
Bob: 3

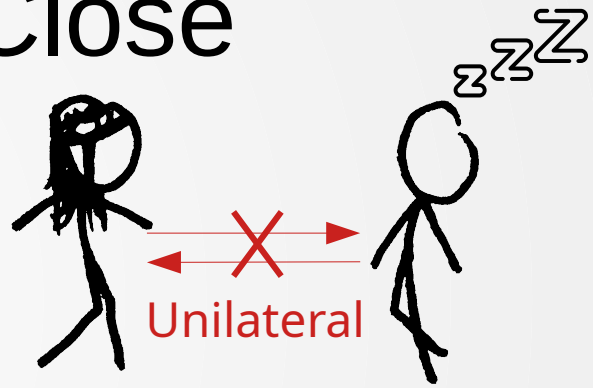
Pay



- Unlimited times
- No touching blockchain

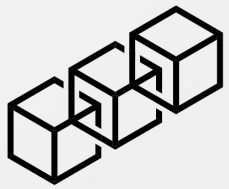


Close



C

Alice: 1.3
Bob: 1.7

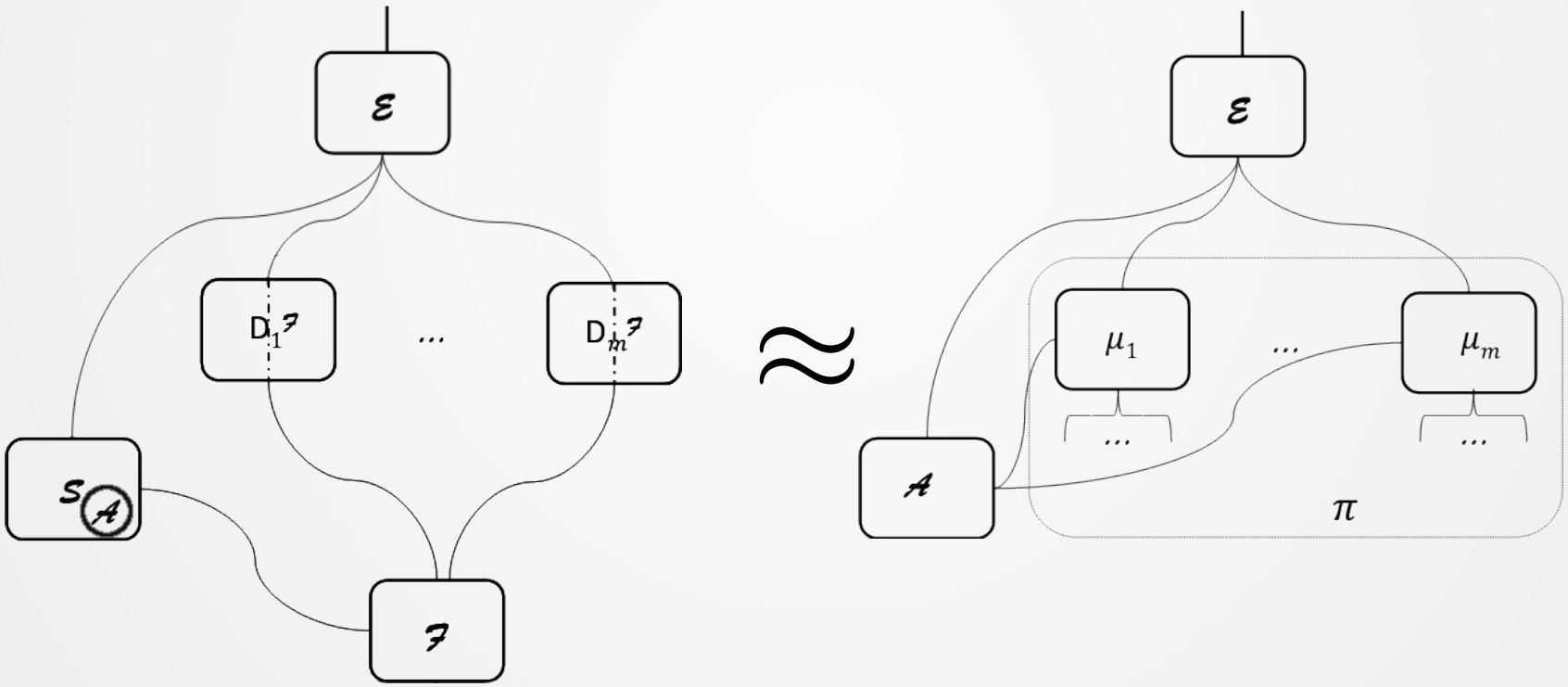


Multi-hop payments

From channels to network!

Simulation-based Security

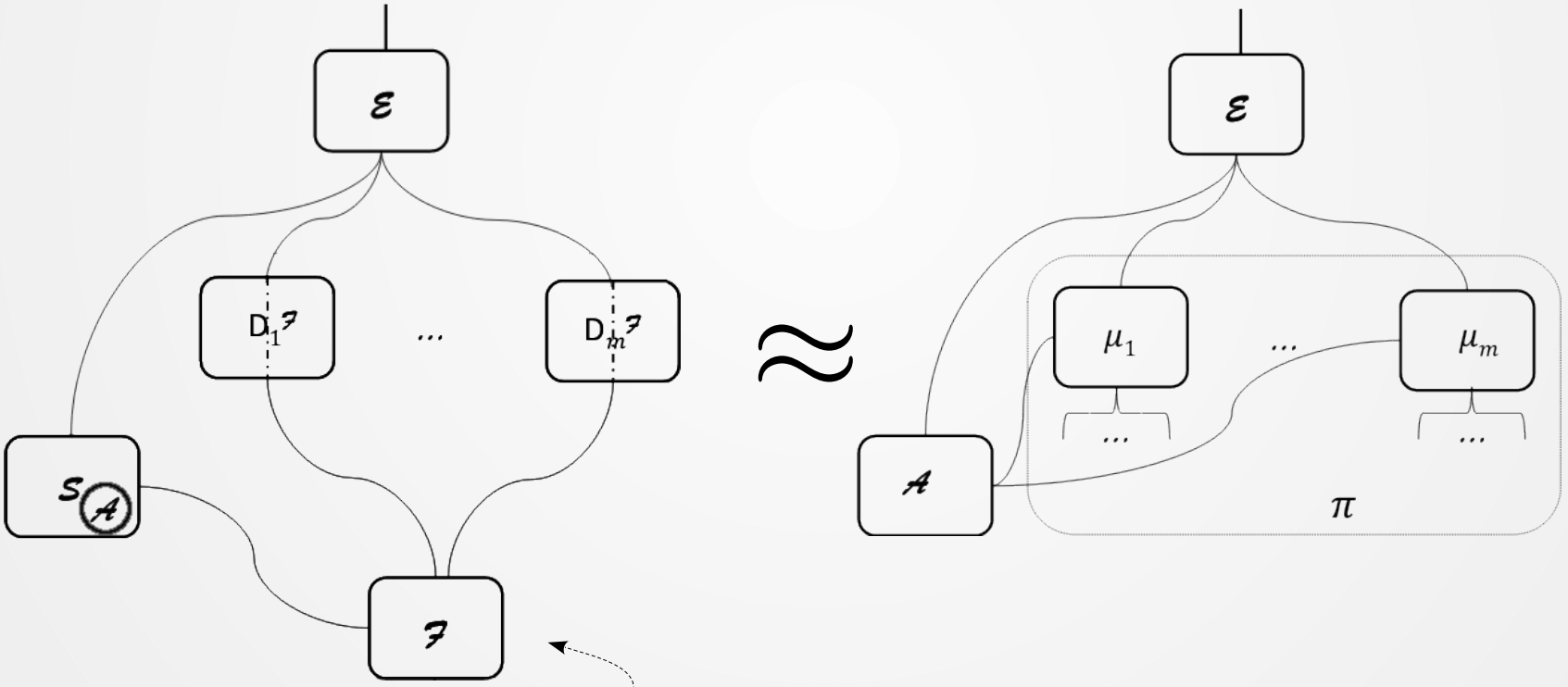
$$\forall \mathcal{E}, \mathcal{A}\ \exists \mathcal{S}:$$



Credits: “Universally Composable Security”, Ran Canetti
https://eprint.iacr.org/2000/067

Our paper

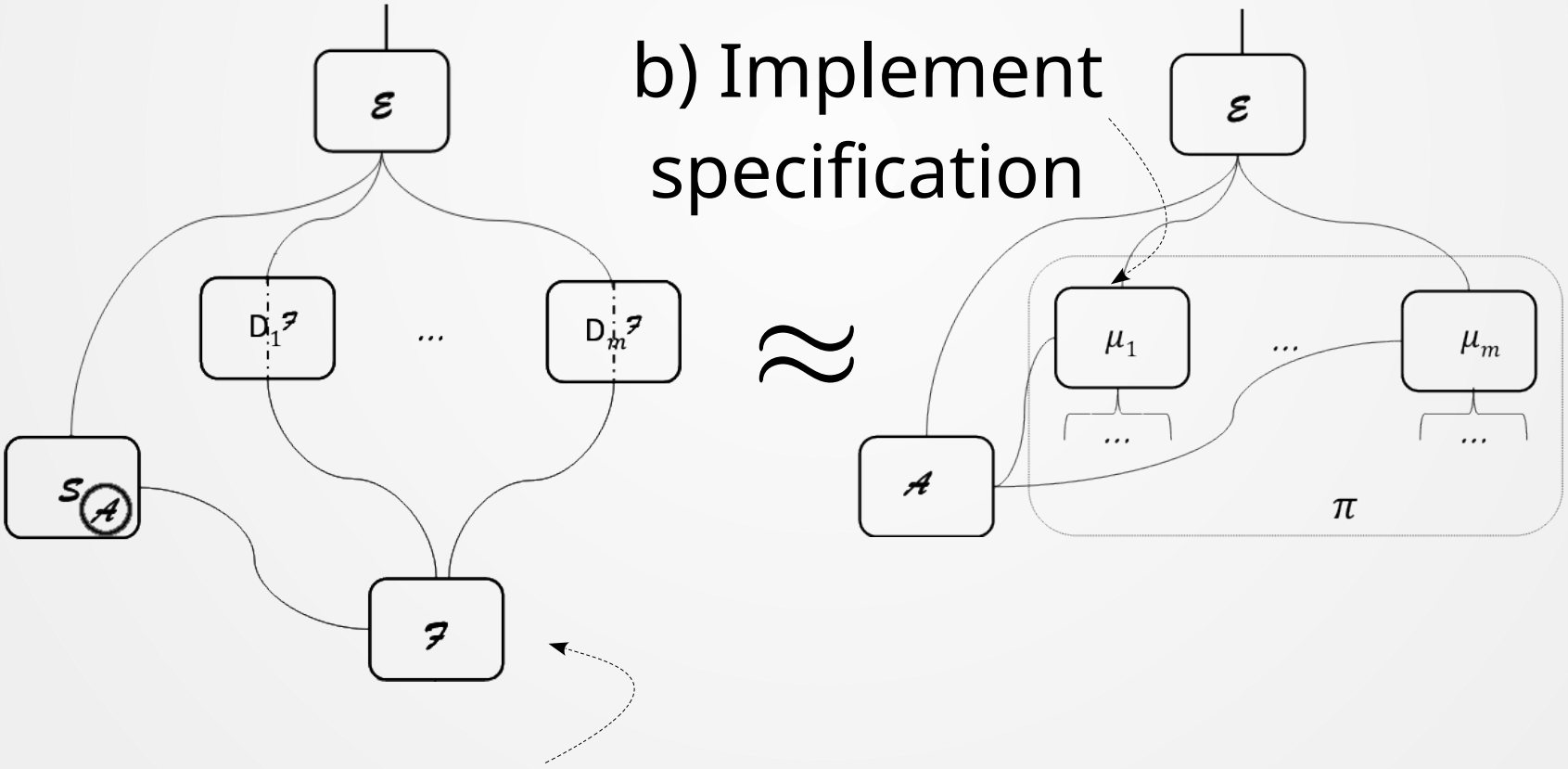
$$\forall \mathcal{E}, \mathcal{A}\ \exists \mathcal{S}:$$



a) Define functionality

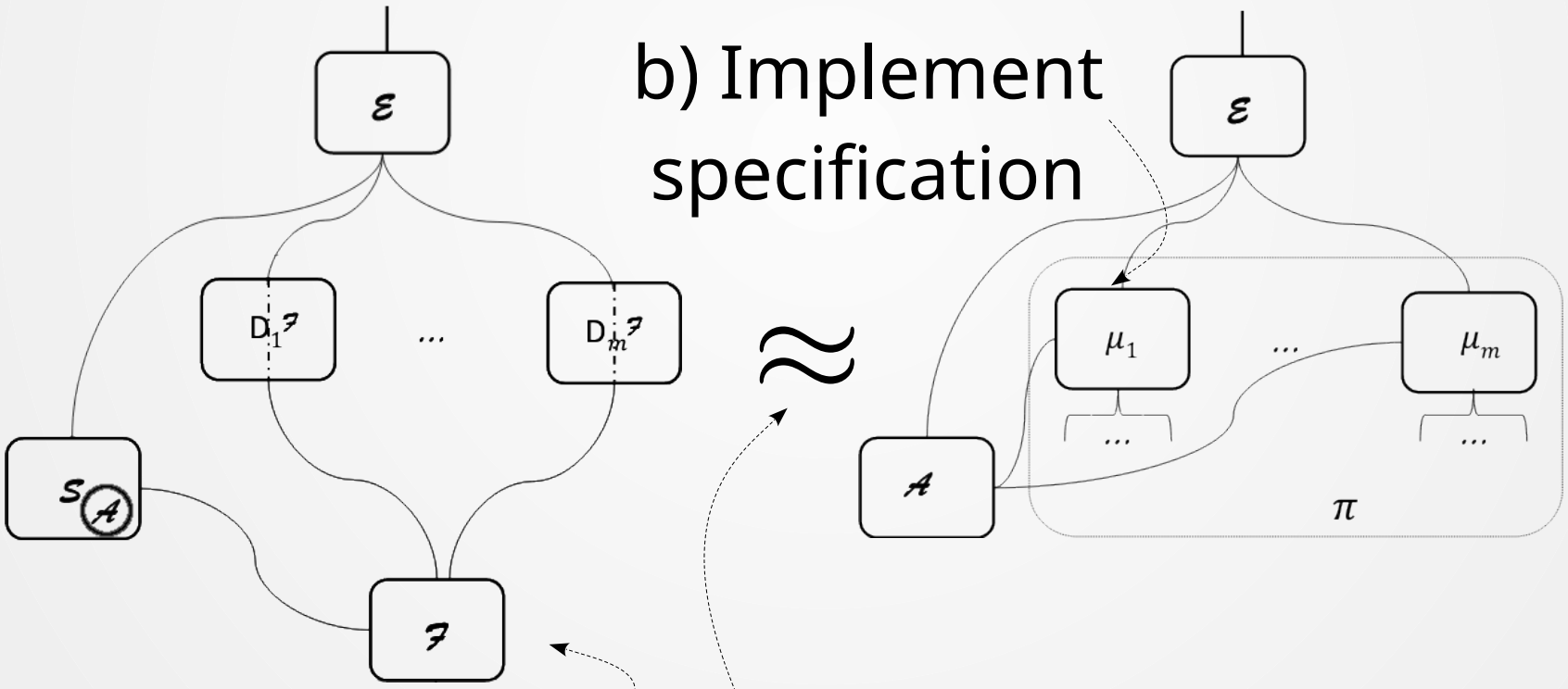
Our paper

$$\forall \mathcal{E}, \mathcal{A}\ \exists \mathcal{S}:$$



Our paper

$$\forall \mathcal{E}, \mathcal{A}\ \exists \mathcal{S}:$$



b) Implement specification

a) Define functionality

c) Prove indistinguishability

Blockchain Functionality

$\mathcal{G}_{\text{ledger}}$ [BMTZ'17, BGKRZ'18]

Functionality

Functionality $\mathcal{F}_{\text{PayNet}}$ – interface

- from $\mathcal{E}$:
  - (REGISTER, delay, relayDelay)
  - (TOPPEDUP)
  - (OPENCHANNEL, *Alice*, *Bob*, x, *tid*)
  - (CHECKFORNEW, *Alice*, *Bob*, *tid*)
  - (PAY, *Bob*, x, $\overrightarrow{\texttt{path}}$, `receipt`)
  - (CLOSECHANNEL, `receipt`, *pchid*)
  - (FORCECLOSECHANNEL, `receipt`, *pchid*)
  - (POLL)
  - (PUSHFULFILL, *pchid*)
  - (PUSHADD, *pchid*)
  - (COMMIT, *pchid*)
  - (FULFILLONCHAIN)
  - (GETNEWS)
- to $\mathcal{E}$:
  - (REGISTER, *Alice*, `delay`(*Alice*), `relayDelay`(*Alice*), pubKey)
  - (REGISTERED)
  - (NEWS, newChannels, closedChannels, updatesToReport)
- from $\mathcal{S}$:
  - (REGISTERDONE, *Alice*, pubKey)
  - (CORRUPTED, *Alice*)
  - (CHANNELANNOUNCED, *Alice*, $p_{Alice,F}$, $p_{Bob,F}$, *fchid*, *pchid*, *tid*)
  - (UPDATE, `receipt`, *Alice*)
  - (CLOSEDCHANNEL, `channel`, *Alice*)
  - (RESOLVEPAYS, *payid*, `charged`)
- to $\mathcal{S}$:
  - (REGISTER, *Alice*, delay, relayDelay)
  - (OPENCHANNEL, *Alice*, *Bob*, x, *fchid*, *tid*)
  - (CHANNELOPENED, *Alice*, *fchid*)
  - (PAY, *Alice*, *Bob*, x, $\overrightarrow{\texttt{path}}$, `receipt`, *payid*)
  - (CONTINUE)
  - (CLOSECHANNEL, *fchid*, *Alice*)
  - (FORCECLOSECHANNEL, *fchid*, *Alice*)
  - (POLL, Σ_{Alice}, *Alice*)
  - (PUSHFULFILL, *pchid*, *Alice*)
  - (PUSHADD, *pchid*, *Alice*)
  - (COMMIT, *pchid*, *Alice*)
  - (FULFILLONCHAIN, t, *Alice*)

How often online?

- No in-flight payments
  - sync at least every `to_self_delay` blocks
- In-flight HTLC – intermediary
  - a = “max new blocks from tx bcast till settled”
  - Sync during [`out_cltv_exp`, `in_cltv_exp` – `2a`]
  - try to publish HTLC-timeout
  - Sync again after `a`
  - If HTLC-success found, update or fulfill on-chain
- In-flight HTLC – payee
  - Fulfill on-chain until `min_final_cltv_expiry` – `a`

Our contributions

- Use a realistic ledger functionality
  - Prove naive ledger unrealizable
- Prove Lightning Network security in UC framework
- Derive exact time bounds for how often parties need to check the chain

Part 2
Recursive Virtual Payment Channels for Bitcoin

Features

- Enables long-lived “virtual” channels w/o an on-chain funding TX
- Virtual channels built on top of a path of preexisting “base” channels – left endpoint funds the channel
- Base channels may themselves be virtual!
- Virtual channels can leverage a path of many base channels

Construction

Intermediary *i* has 3 classes of TXs:

- “Initiator” TX
  - Spends left & right funding outputs
  - Has virtual output with interval [*i*]

Construction

Intermediary *i* has 3 classes of TXs:

- “Initiator” TX
  - Spends left & right funding outputs
  - Has virtual output with interval [*i*]
- “Extend-interval” TXs
  - Spends 1 funding output and 1 virtual output with interval [*j*, …, *i*-1] or [*i*+1, …, *j*]
  - Has virtual output w/ interval [*j*, …, *i*] or [*i*, …, *j*]

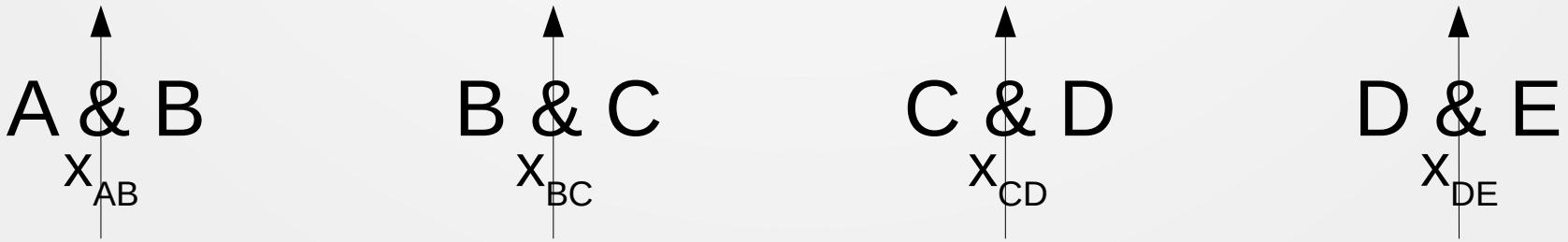
Construction

Intermediary *i* has 3 classes of TXs:

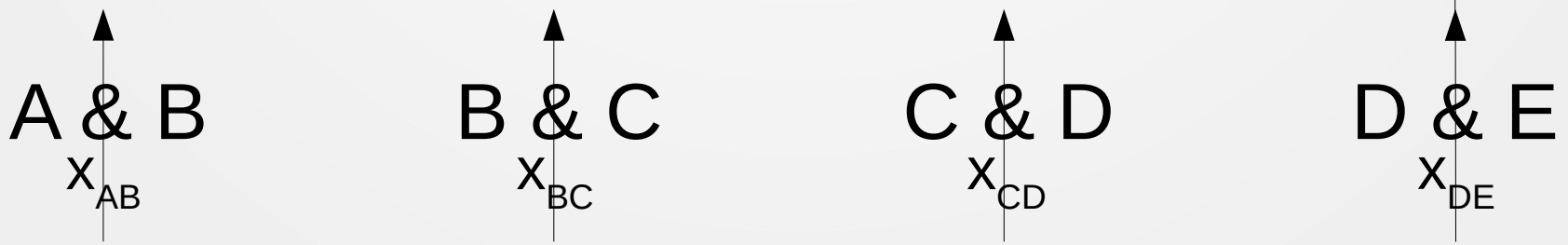
- "Initiator" TX
  - Spends left & right funding outputs
  - Has virtual output with interval [*i*]
- "Extend-interval" TXs
  - Spends 1 funding output and 1 virtual output with interval [*j*, ..., *i*-1] or [*i*+1, ..., *j*]
  - Has virtual output w/ interval [*j*, ..., *i*] or [*i*, ..., *j*]
- "Merge-intervals" TXs
  - Spends 2 virtual outputs with intervals [*j*, ..., *i*-1] and [*i*+1, ..., k]
  - Has virtual output with interval [*j*, ..., *k*]

Example 1
Fundee wants to close

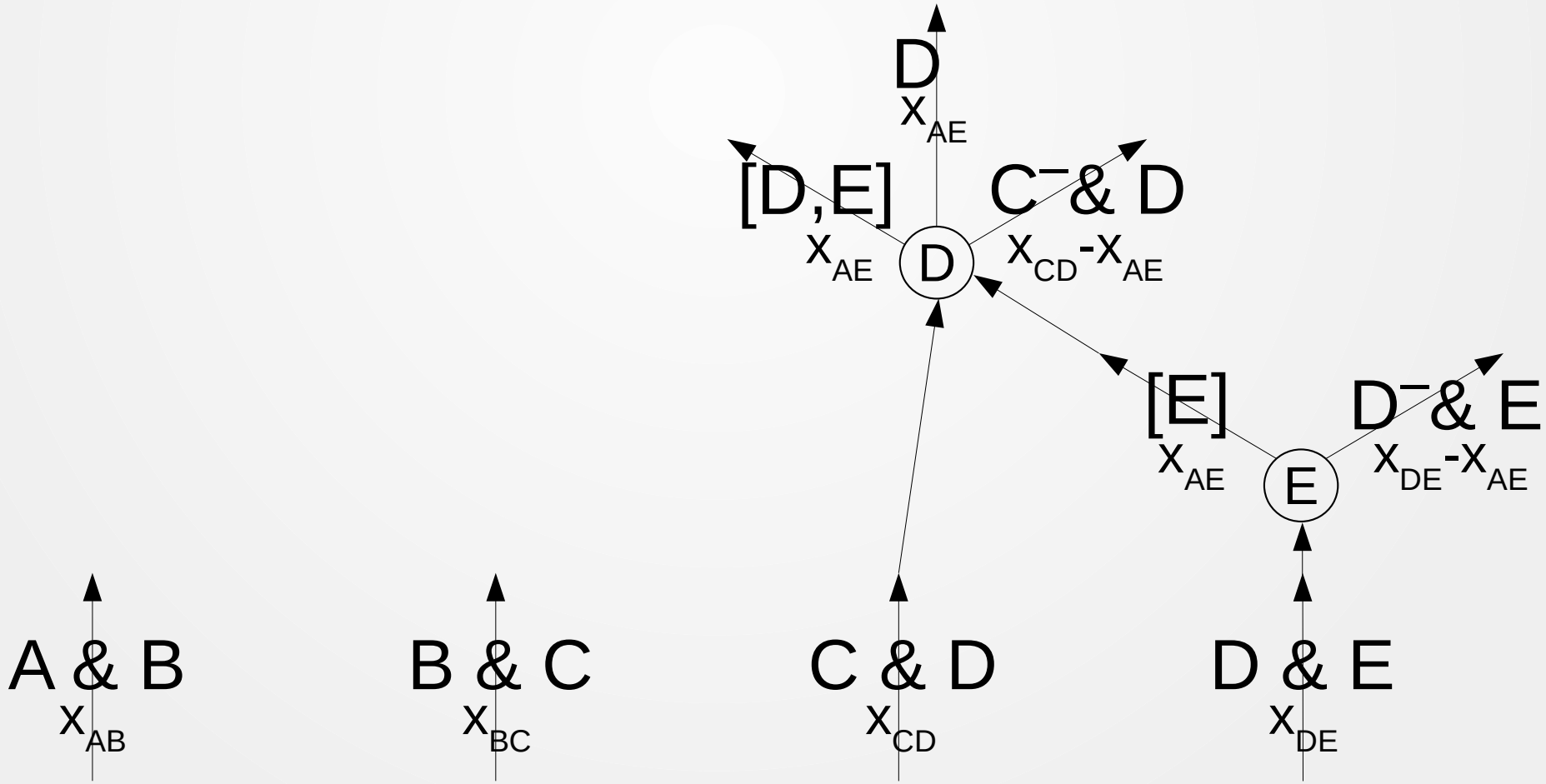
t=0

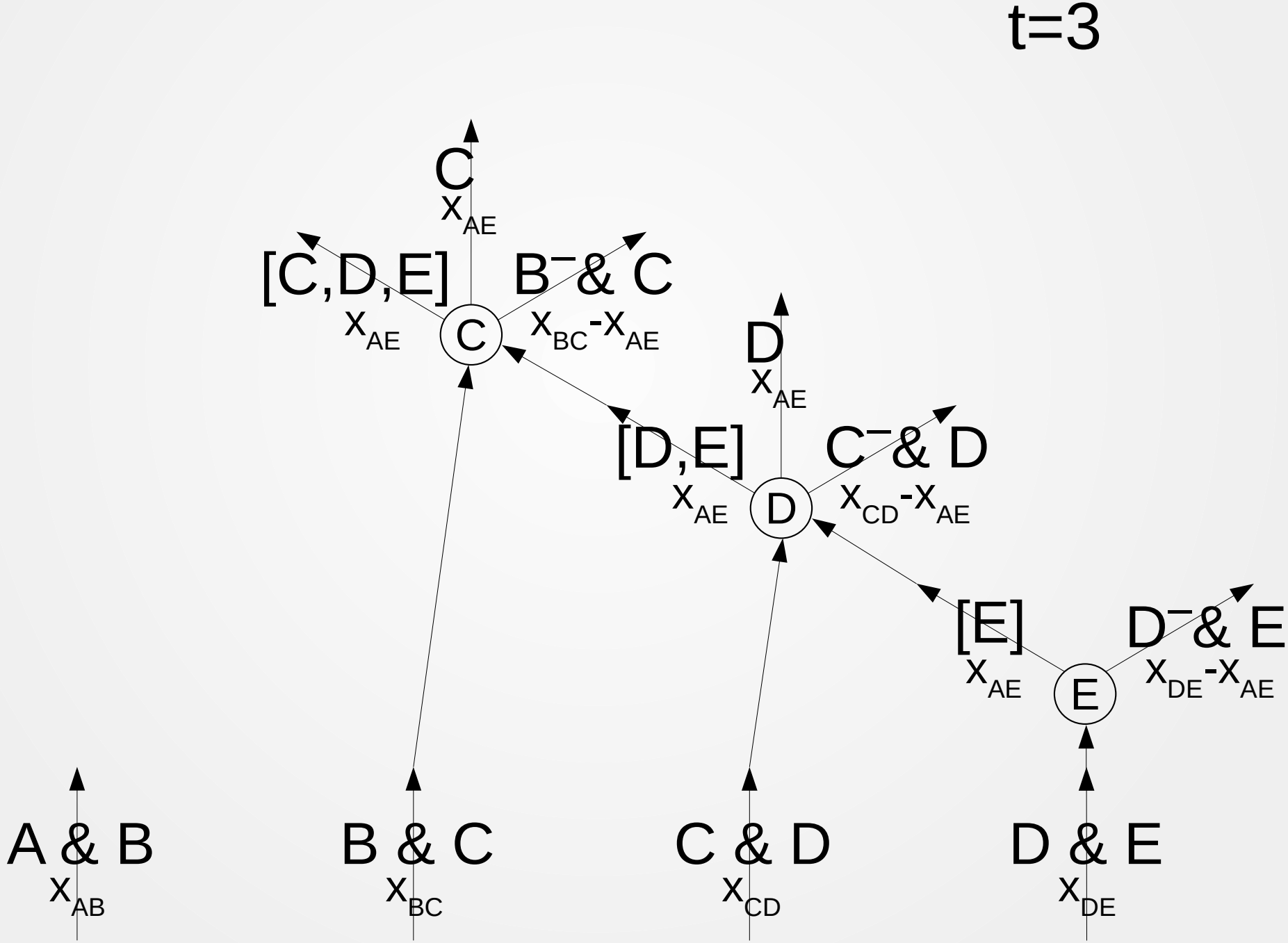


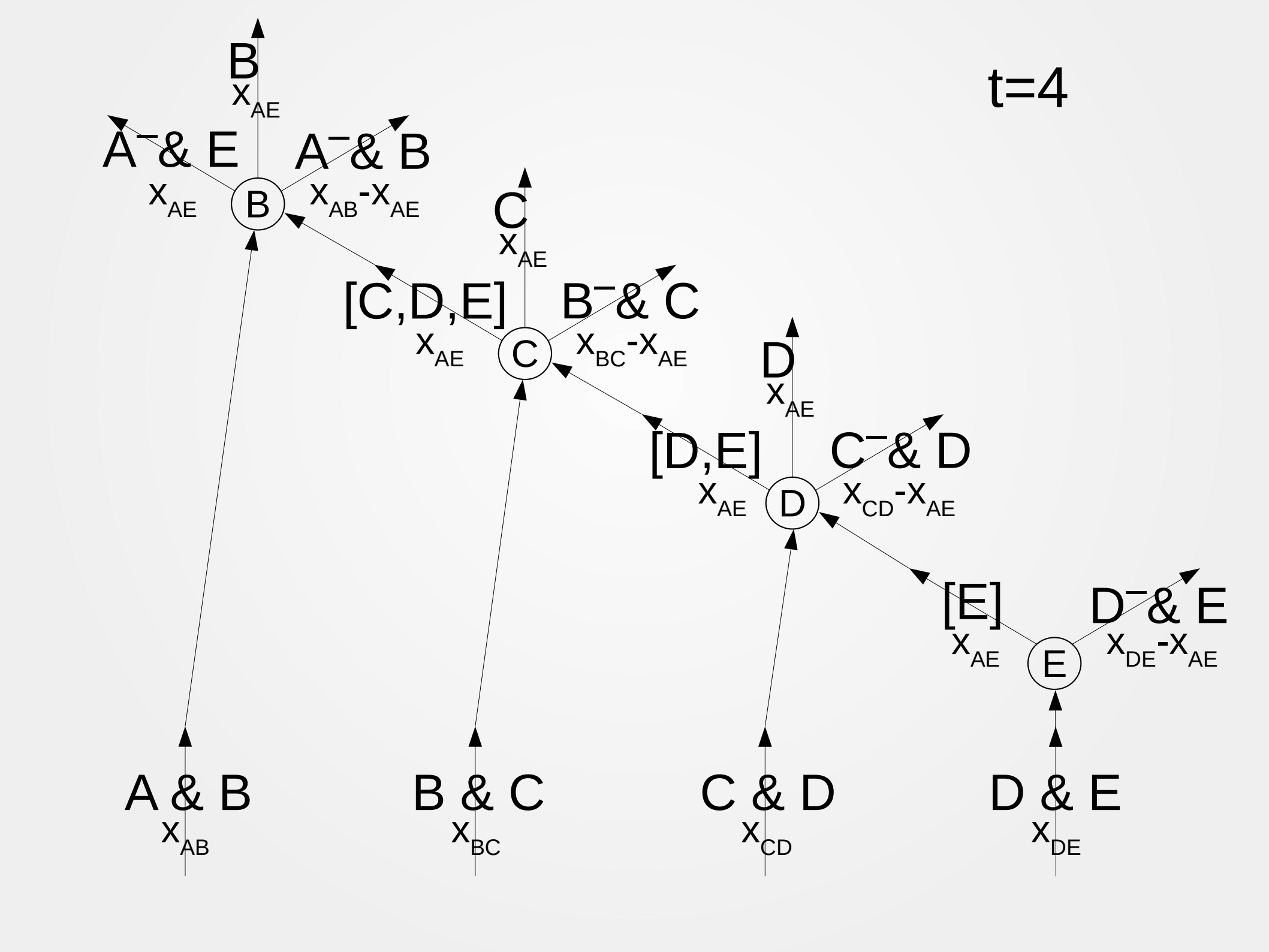
t=1



t=2

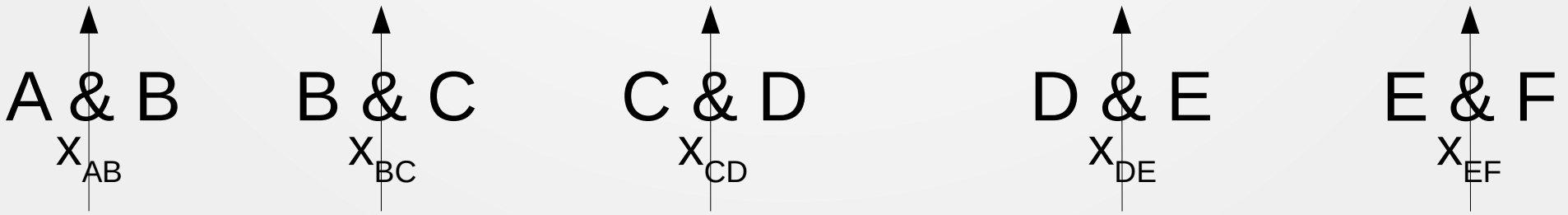

D
x_{AE}
[D,E]
x_{AE}
D
$C^- \& D$
x_{CD}-x_{AE}
[E]
x_{AE}
E
$D^- \& E$
x_{DE}-x_{AE}
A & B
x_{AB}
B & C
x_{BC}
C & D
x_{CD}
D & E
x_{DE}


t=3
C
x_AE
[C,D,E]
x_AE
C
B⁻ & C
x_BC-x_AE
D
x_AE
[D,E]
x_AE
D
C⁻ & D
x_CD-x_AE
[E]
x_AE
E
D⁻ & E
x_DE-x_AE
A & B
x_AB
B & C
x_BC
C & D
x_CD
D & E
x_DE

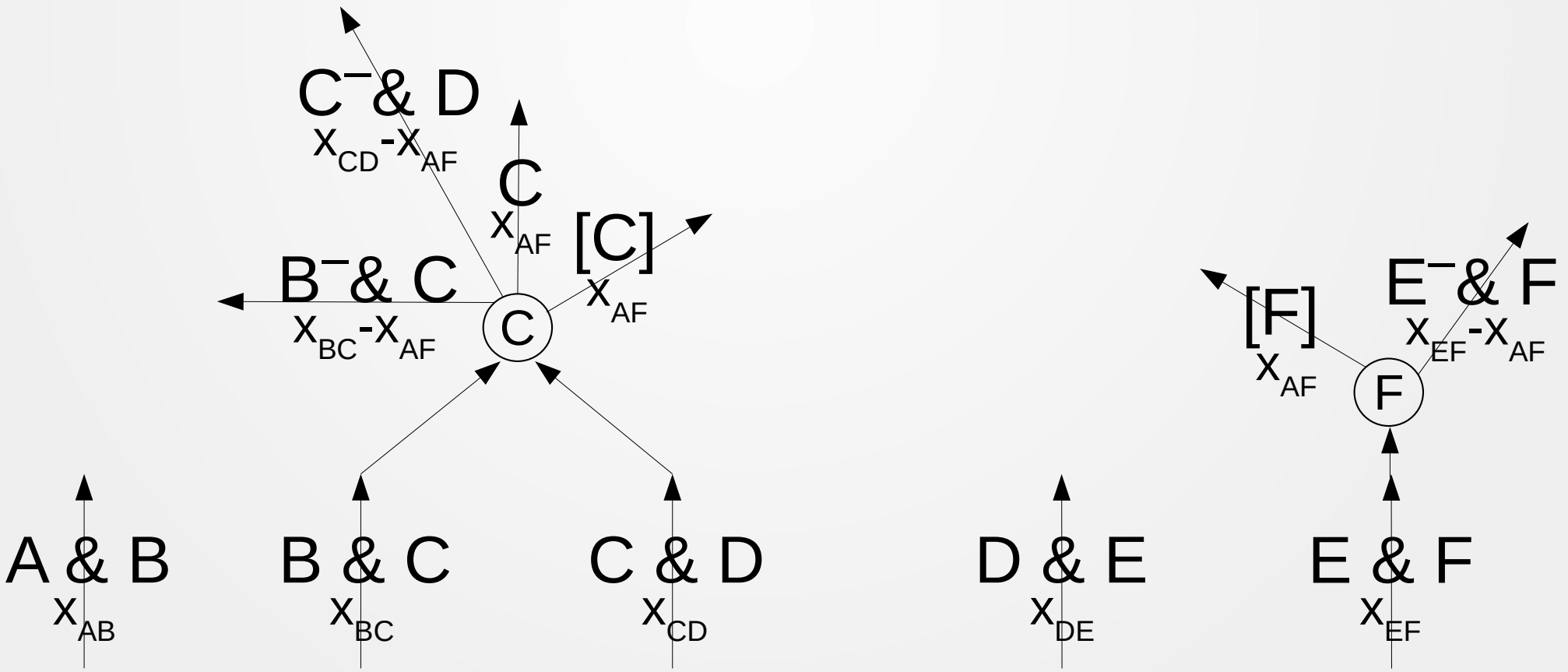

t=4
B
x_{AE}
$A^- \& E$
x_{AE}
B
$A^- \& B$
$x_{AB}-x_{AE}$
C
x_{AE}
[C,D,E]
x_{AE}
C
$B^- \& C$
$x_{BC}-x_{AE}$
D
x_{AE}
[D,E]
x_{AE}
D
$C^- \& D$
$x_{CD}-x_{AE}$
[E]
x_{AE}
E
$D^- \& E$
$x_{DE}-x_{AE}$
A & B
x_{AB}
B & C
x_{BC}
C & D
x_{CD}
D & E
x_{DE}

Example 2
Simultaneous initiators

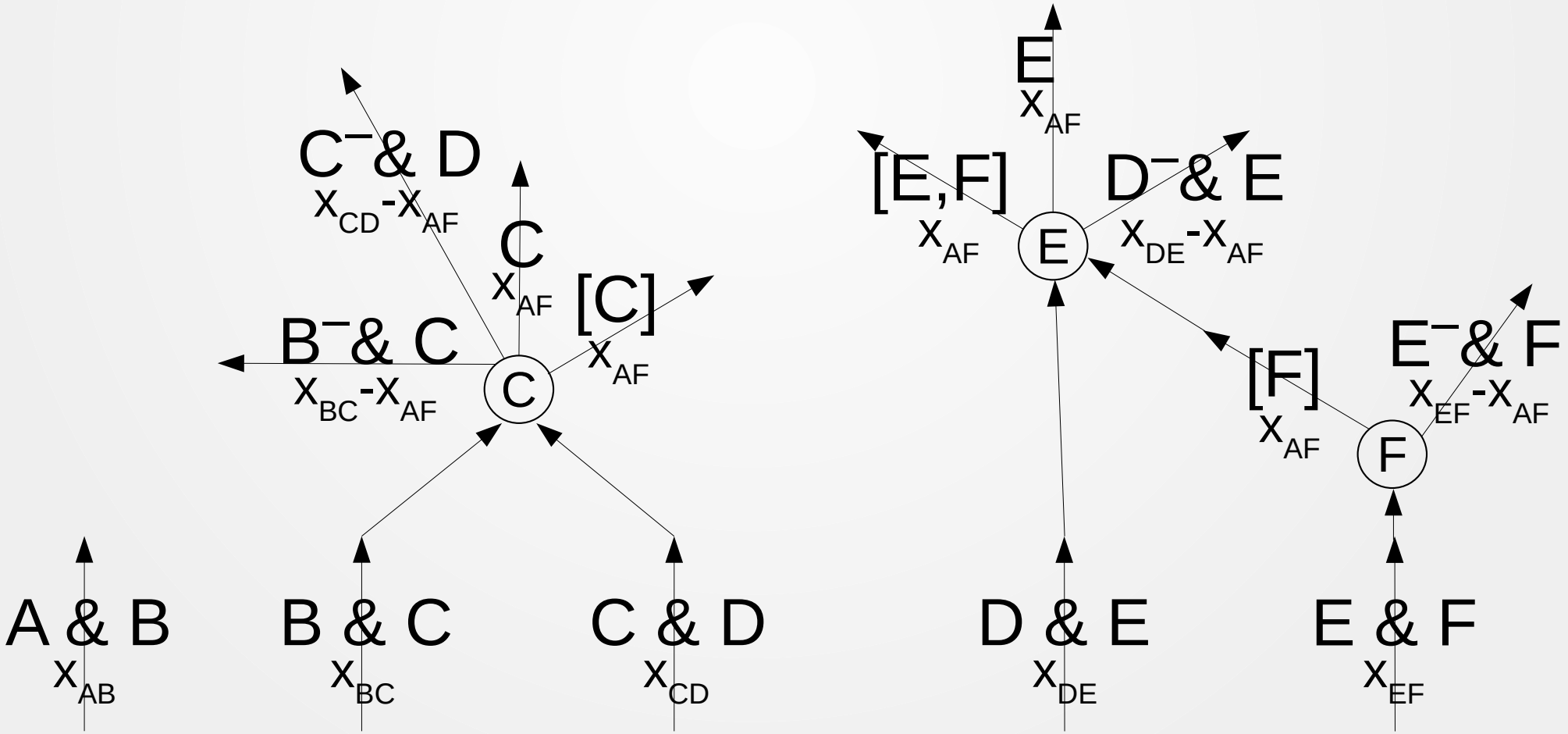
t=0



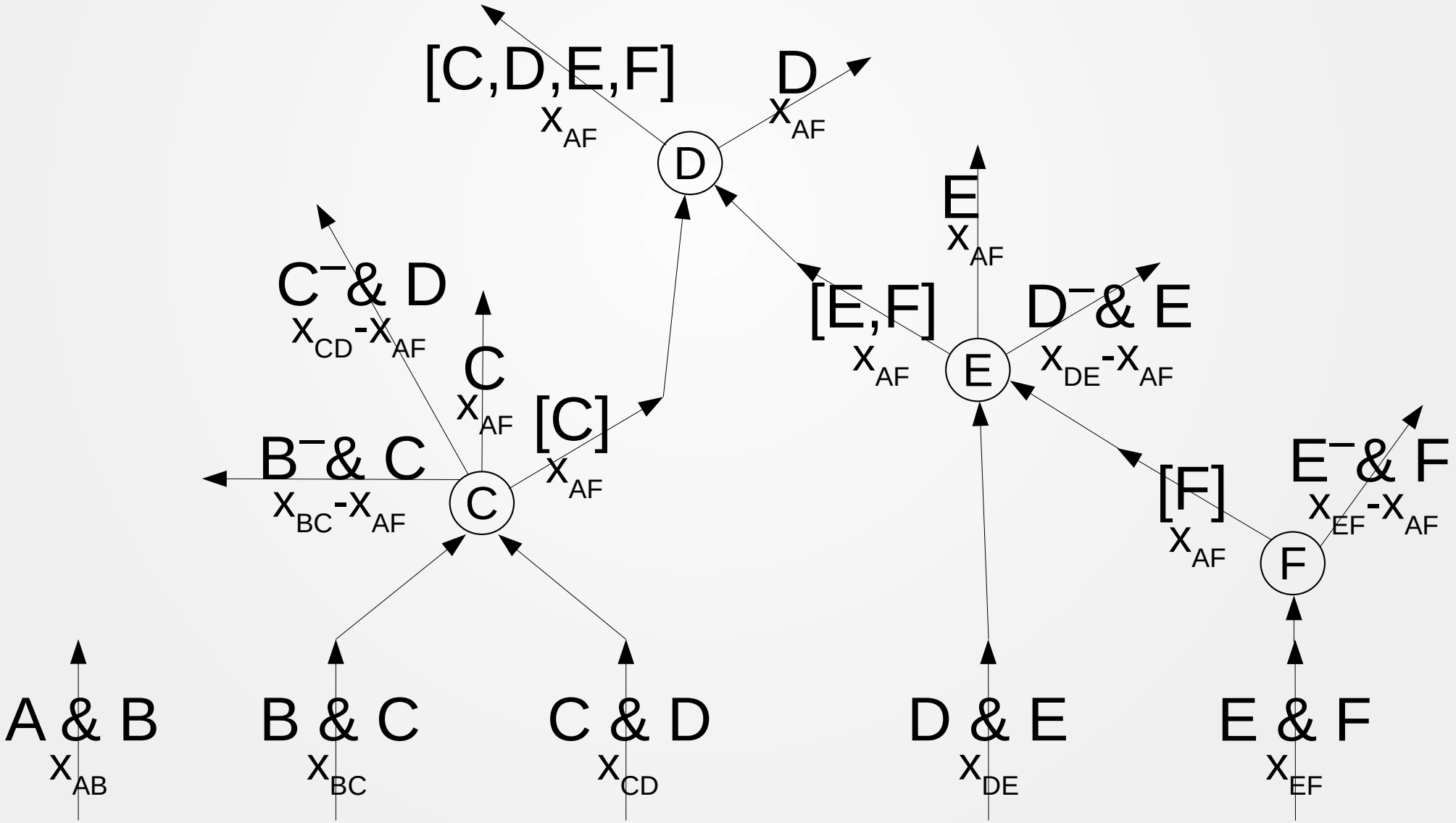
t=1


C⁻ & D
x_CD-x_AF
C
x_AF
[C]
x_AF
B⁻ & C
x_BC-x_AF
C
A & B
x_AB
B & C
x_BC
C & D
x_CD
D & E
x_DE
[F]
x_AF
E⁻ & F
x_EF-x_AF
F
E & F
x_EF

t=2

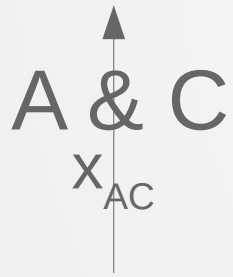

A & B
x_{AB}
B & C
x_{BC}
C & D
x_{CD}
C
x_{AF}
C^{-}& D
x_{CD}-x_{AF}
B^{-}& C
x_{BC}-x_{AF}
[C]
x_{AF}
C
D & E
x_{DE}
E & F
x_{EF}
E
x_{AF}
[E,F]
x_{AF}
D^{-}& E
x_{DE}-x_{AF}
E
[F]
x_{AF}
E^{-}& F
x_{EF}-x_{AF}
F

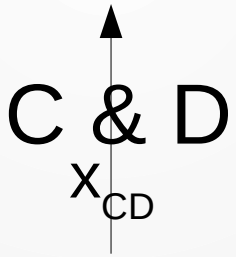
t=3

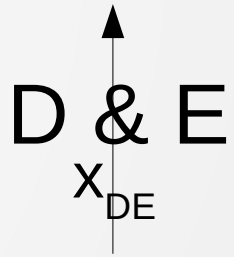

[C,D,E,F]
x_{AF}
D
x_{AF}
D
E
x_{AF}
C^{-}& D
x_{CD}-x_{AF}
C
x_{AF}
[C]
x_{AF}
[E,F]
x_{AF}
E
D^{-}& E
x_{DE}-x_{AF}
B^{-}& C
x_{BC}-x_{AF}
C
[F]
x_{AF}
E^{-}& F
x_{EF}-x_{AF}
F
A & B
x_{AB}
B & C
x_{BC}
C & D
x_{CD}
D & E
x_{DE}
E & F
x_{EF}

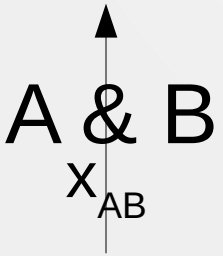
Example 3
Virtual base channel

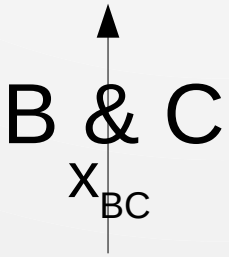
t=0



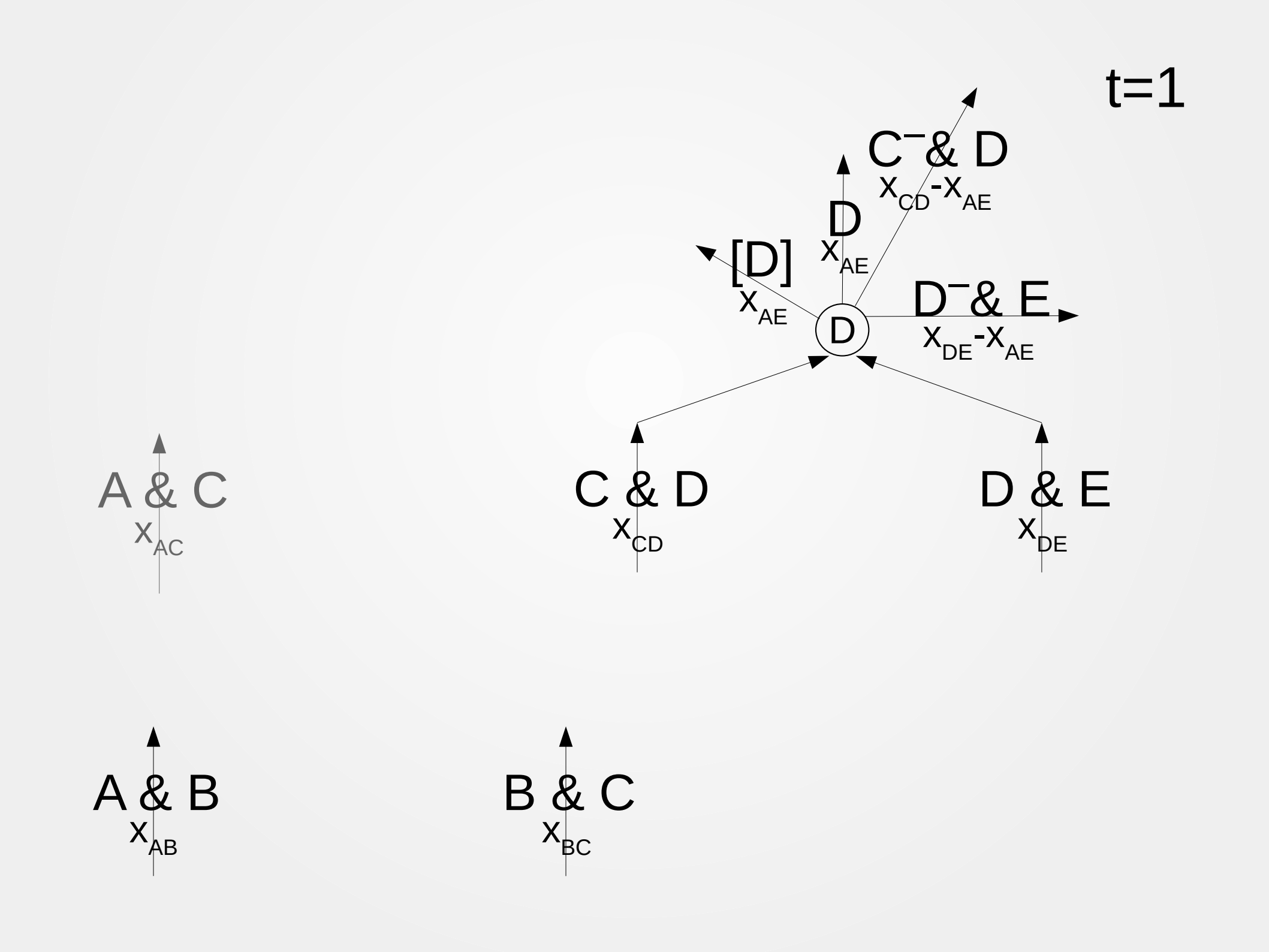




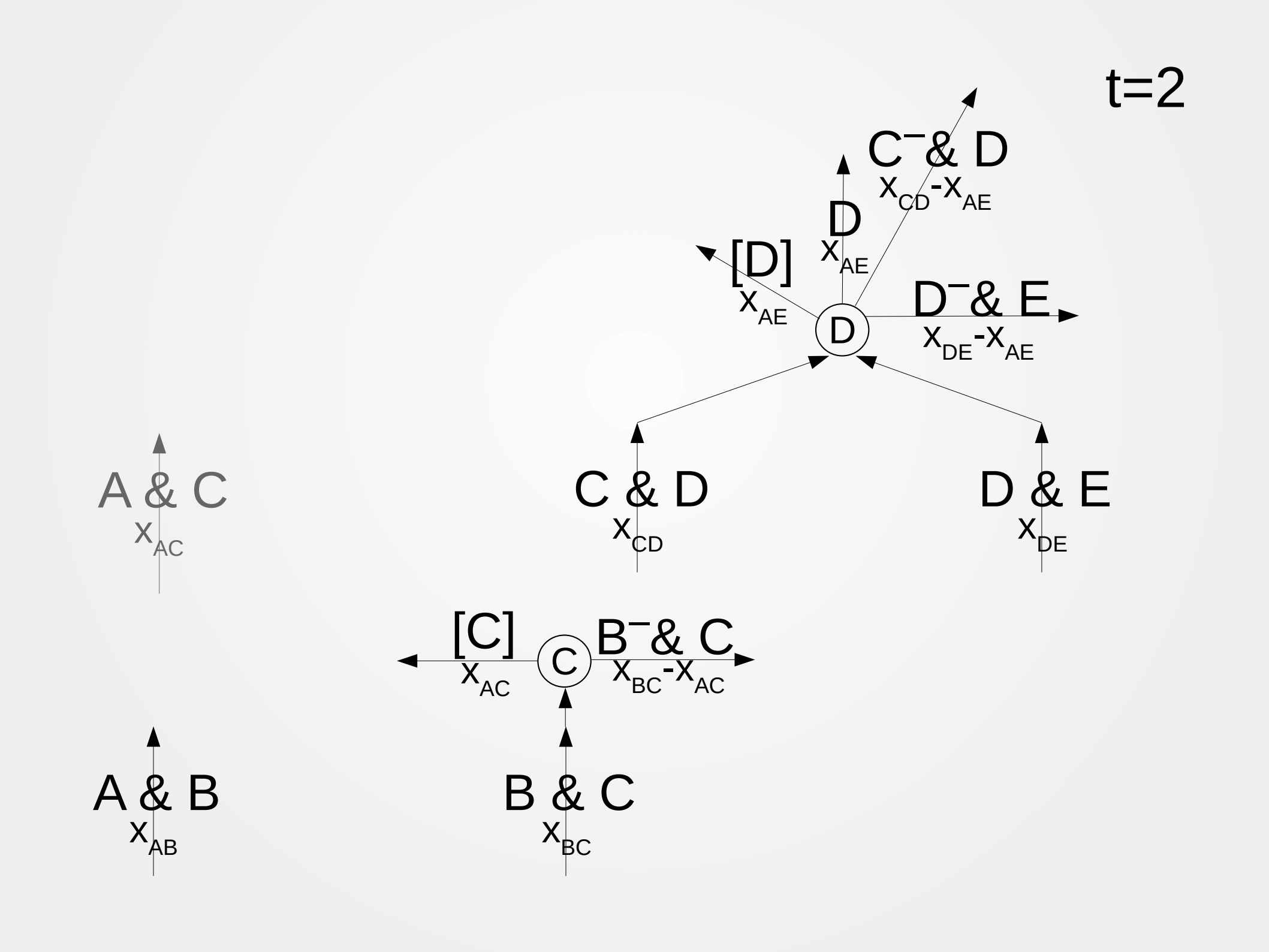


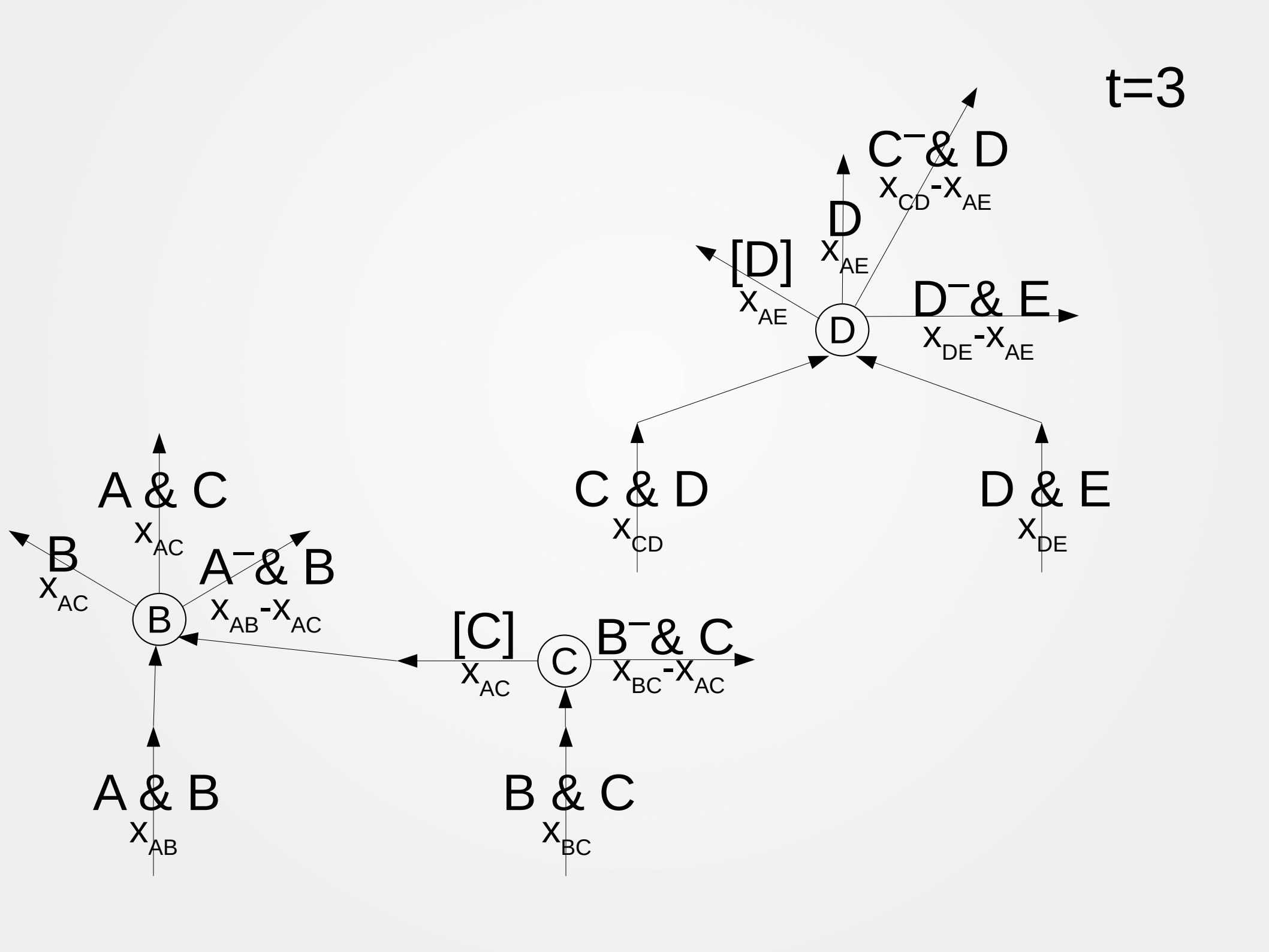


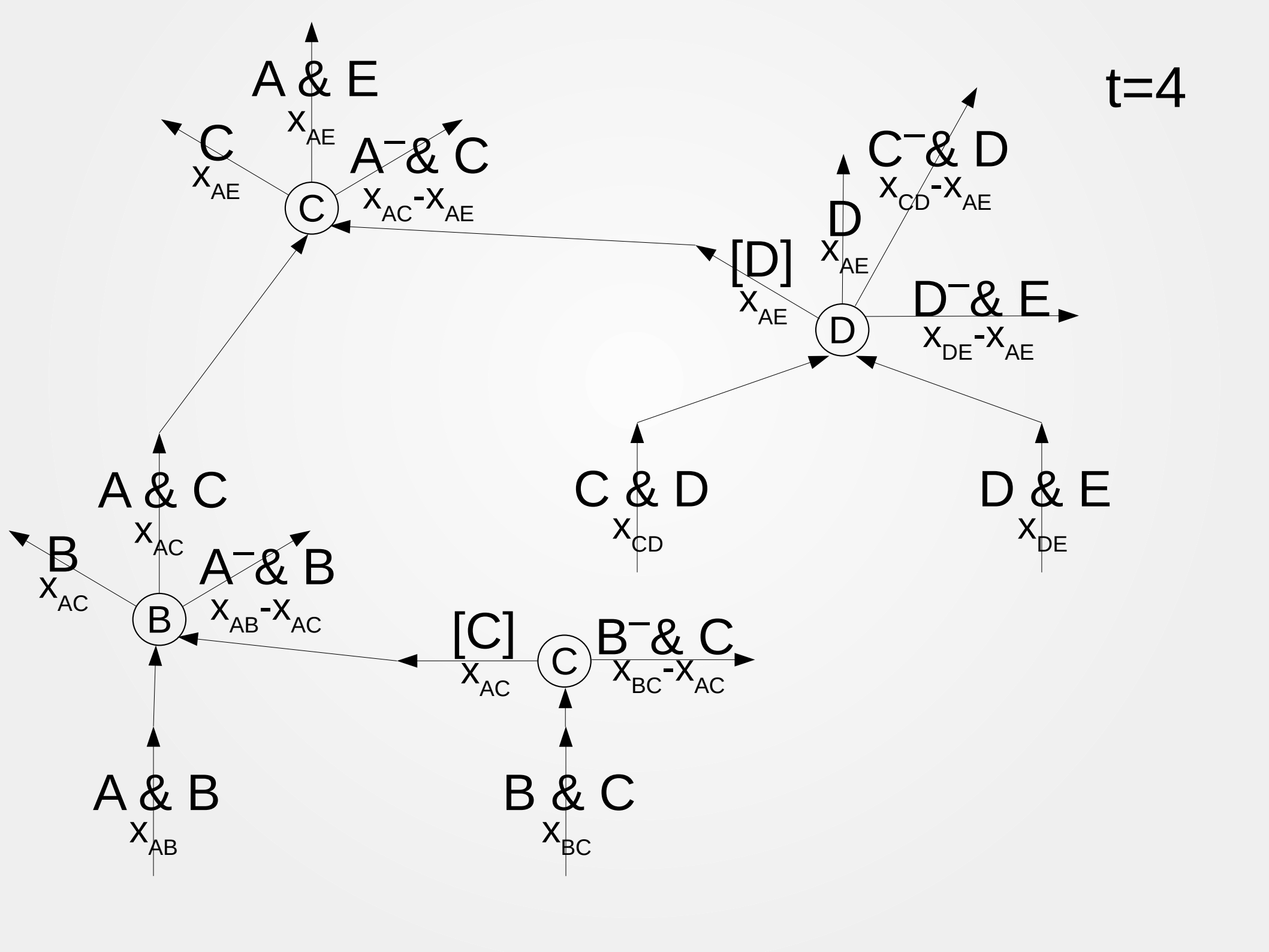
t=1


C⁻ & D
x_CD -x_AE
D
x_AE
[D]
x_AE
D⁻ & E
x_DE -x_AE
D
A & C
x_AC
C & D
x_CD
D & E
x_DE
A & B
x_AB
B & C
x_BC

t=2


C⁻ & D
x_CD - x_AE
D
x_AE
[D]
x_AE
D
D⁻ & E
x_DE - x_AE
A & C
x_AC
C & D
x_CD
D & E
x_DE
[C]
x_AC
C
B⁻ & C
x_BC - x_AC
A & B
x_AB
B & C
x_BC


t=3
C⁻ & D
x_CD - x_AE
D
x_AE
[D]
x_AE
D⁻ & E
x_DE - x_AE
D
C & D
x_CD
D & E
x_DE
A & C
x_AC
B
x_AC
A⁻ & B
x_AB - x_AC
B
[C]
x_AC
C
B⁻ & C
x_BC - x_AC
A & B
x_AB
B & C
x_BC


t=4
A & E
x_AE
C
x_AE
A⁻ & C
x_AC-x_AE
C
[D]
x_AE
C⁻ & D
x_CD-x_AE
D
x_AE
D⁻ & E
x_DE-x_AE
D
A & C
x_AC
C & D
x_CD
D & E
x_DE
B
x_AC
A⁻ & B
x_AB-x_AC
B
[C]
x_AC
B⁻ & C
x_BC-x_AC
C
A & B
x_AB
B & C
x_BC

Design decisions

- UC secure
- Intuitive functionality
- State machine
- Uses $\mathcal{G}_{\mathrm{ledger}}$

Summary

- Composable analysis of the Lightning Network
- Construction and composable analysis of Recursive Virtual Channels for Bitcoin

Thank you!