A Guide to Risk Limiting Audits for Instant Runoff Voting (IRV)

Vanessa Teague[1] Michelle Blom[1,2] Peter Stuckey[3]

Andrew Conway Mark Pinoli[1]

[1] Democracy Developers Ltd.

[2] School of Computing and Information Systems, The University of Melbourne

[3] Faculty of IT, Monash University

© Democracy Developers Ltd.

RLAs for IRV: How did we get here?

- All Australian elections are Ranked Choice
- We wanted to bring RLAs to Aus
- We developed algorithms for RLA IRVs
- We piloted them in San Francisco in 2019
  - Also some party primaries

RLAs for IRV: How did we get here?

You can do RLAs for IRV

The Process Pilot of Risk-Limiting Audits for the San Francisco District Attorney 2019 Instant Runoff Vote

Michelle Blom[*], Andrew Conway[†], Dan King[‡], Laurent Sandrolini[§], Philip B. Stark[¶], Peter J. Stuckey[||] and Vanessa Teague[**]

April 2, 2020

The City and County of San Francisco, CA, has used Instant Runoff Voting (IRV) for some elections since 2004. This report describes the first ever process pilot of Risk Limiting Audits for IRV, for the San Francisco District Attorney's race in November, 2019. We found that the vote-by-mail outcome could be efficiently audited to well under the 0.05 risk limit given a sample of only 200 ballots. All the software we developed for the pilot is open source.

RLAs for IRV: Concepts

RLA BASICS

HOW THEY APPLY TO IRV

TREE STRUCTURES

VISUALIZING IRV OUTCOMES

(WITH TREES!)

ASSERTIONS

AUDITING ASSERTIONS

RLA Basics and How they apply to IRV

What is a Risk Limiting Audit?

- A post-election activity to check **the right person won**
- Involve randomly sampling paper ballots
- Compute statistics on this sample to ascertain a level of **risk**
- An RLA guarantees a **risk limit** – the maximum probability that it will mistakenly confirm a reported outcome when it was in fact wrong
- Ballots are sampled until this risk falls below an acceptable level

What is Instant Runoff Voting?

A form of ranked-vote or preferential voting

While there is more than one continuing candidate[1]:

From the continuing candidates, select the candidate *C* with the smallest tally

Eliminate *C*:

Give each ballot in *C*'s tally to the next-preferred continuing candidate on that ballot

[1] or while no continuing candidate has the majority of votes

What is an IRV RLA?

What does it do?

Checks that the announced winner won.

What does it not do?

- Check whether they won by the announced elimination order
- Check the IRV tabulations
- Check the runner-up is correct

This is a deliberate design feature: RAIRE does not waste auditing effort on details that do not affect who won.

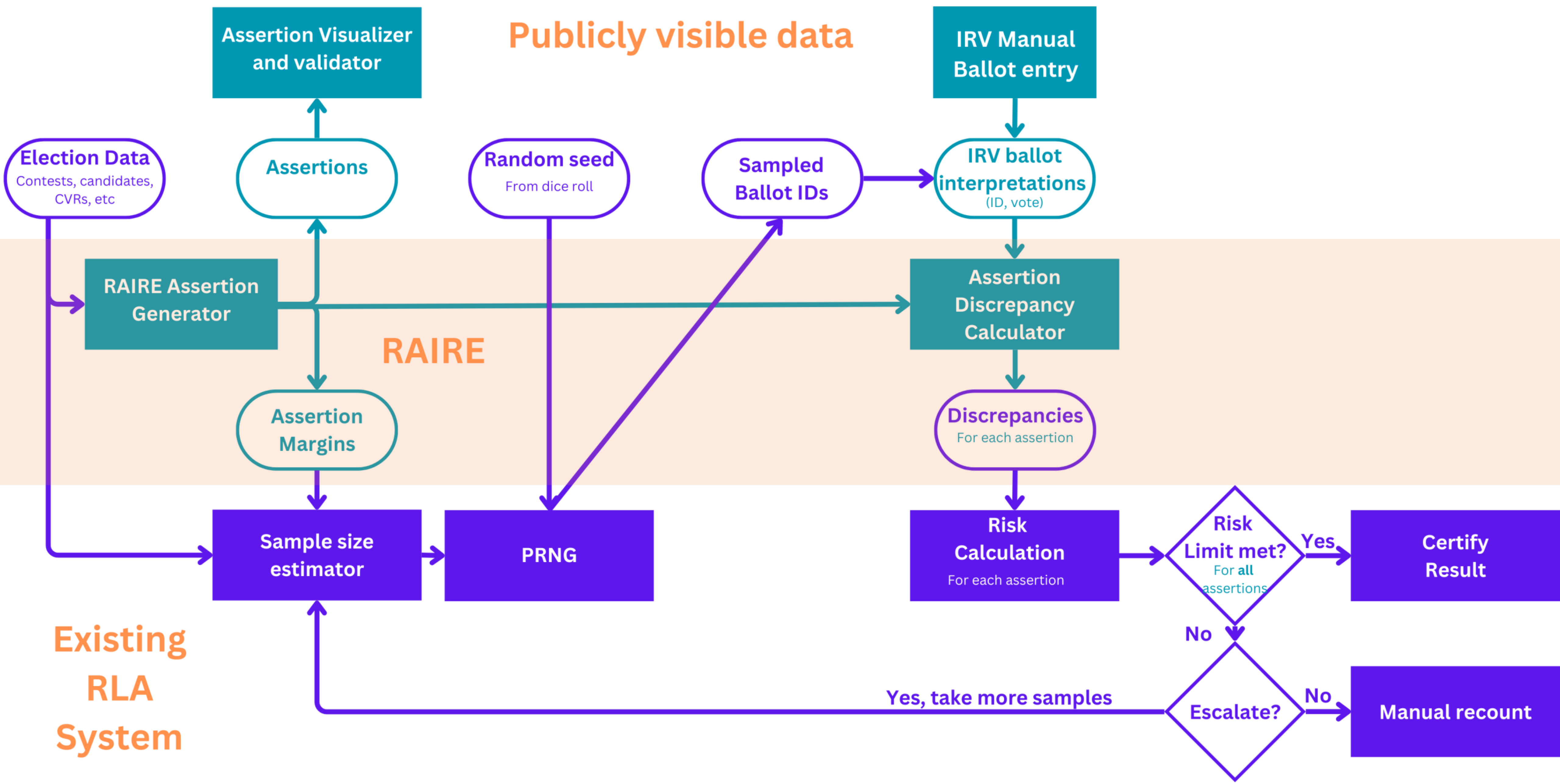
What is RAIRE?

What does it stand for?

R*isk Limiting* ***A****udits for* ***IR****V* ***E****lections*

How do you audit?

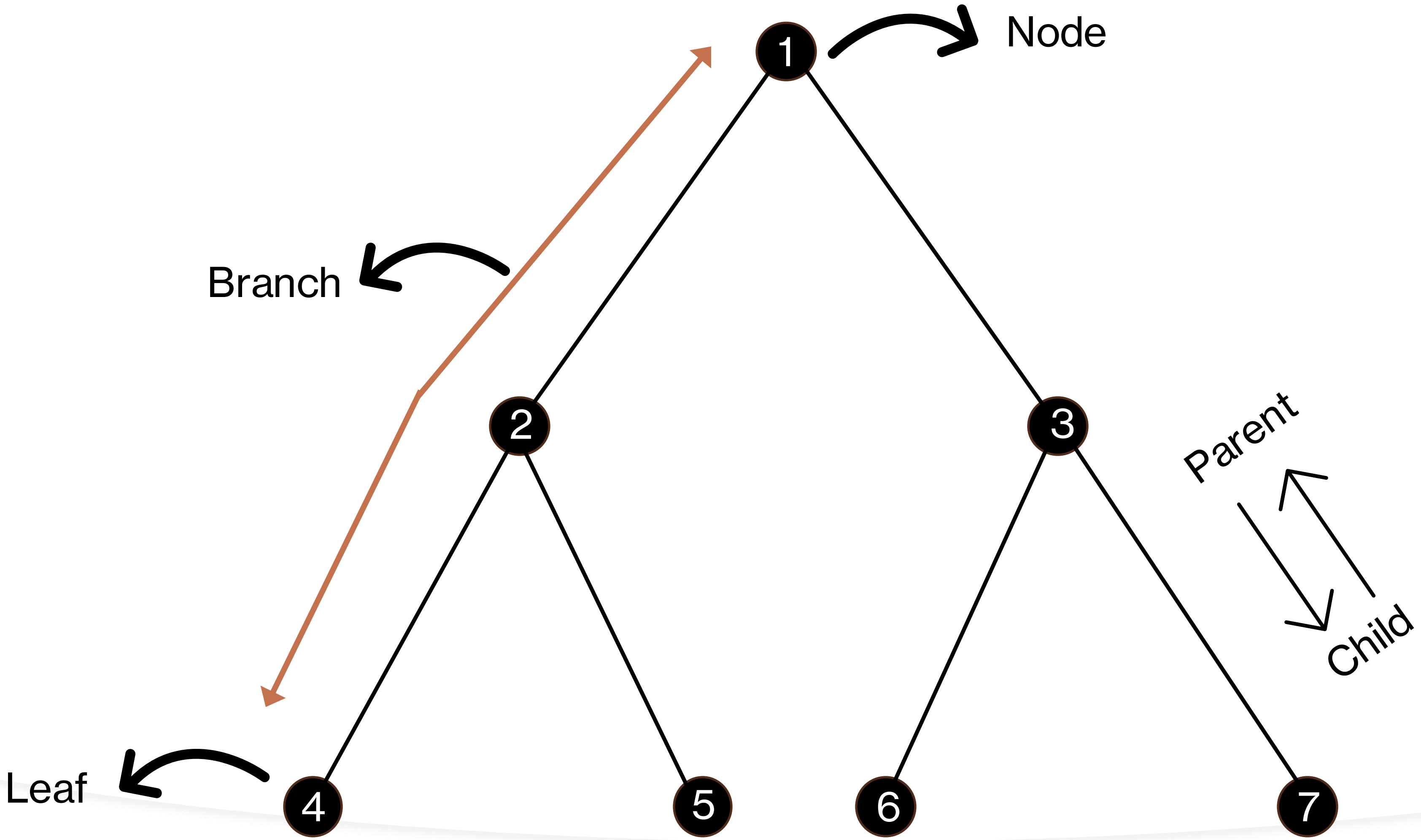
1. RAIRE generates a set of Assertions that imply that the announced winner won.
2. These Assertions are tested with an RLA
    - hence making an RLA of the IRV election result.



Publicly visible data
Assertion Visualizer and validator
IRV Manual Ballot entry
Election Data
Contents, candidates, CVRs, etc
Assertions
Random seed
From dice roll
Sampled Ballot IDs
IRV ballot interpretations
(ID, vote)
RAIRE Assertion Generator
Assertion Discrepancy Calculator
RAIRE
Assertion Margins
Discrepancies
For each assertion
Sample size estimator
PRNG
Risk Calculation
For each assertion
Risk Limit met?
For all assertions
Yes
Certify Result
No
Existing RLA System
Yes, take more samples
Escalate?
No
Manual recount

Tree Structures

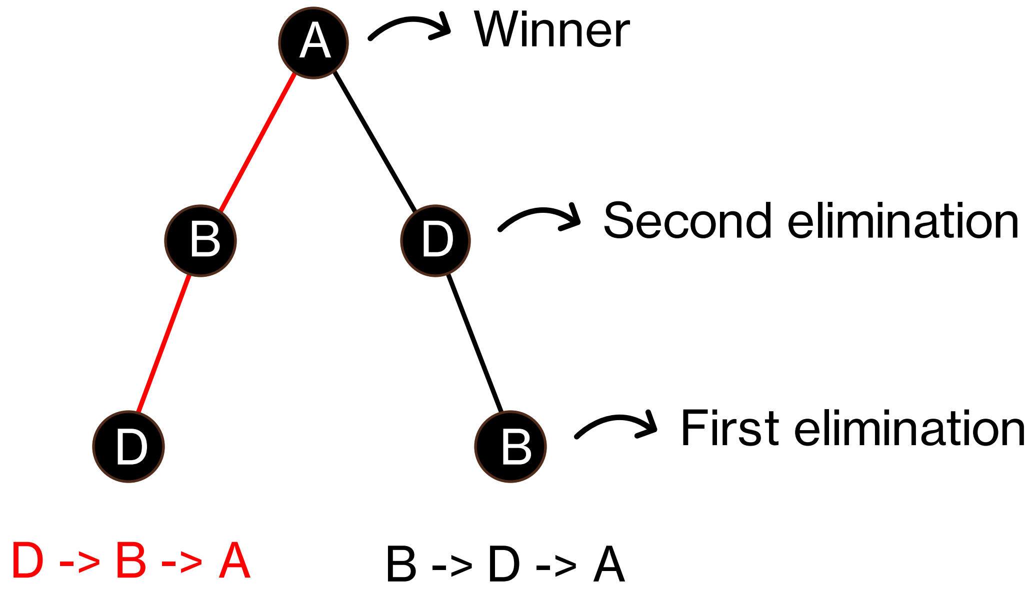
Tree Structures



Visualizing IRV Outcomes

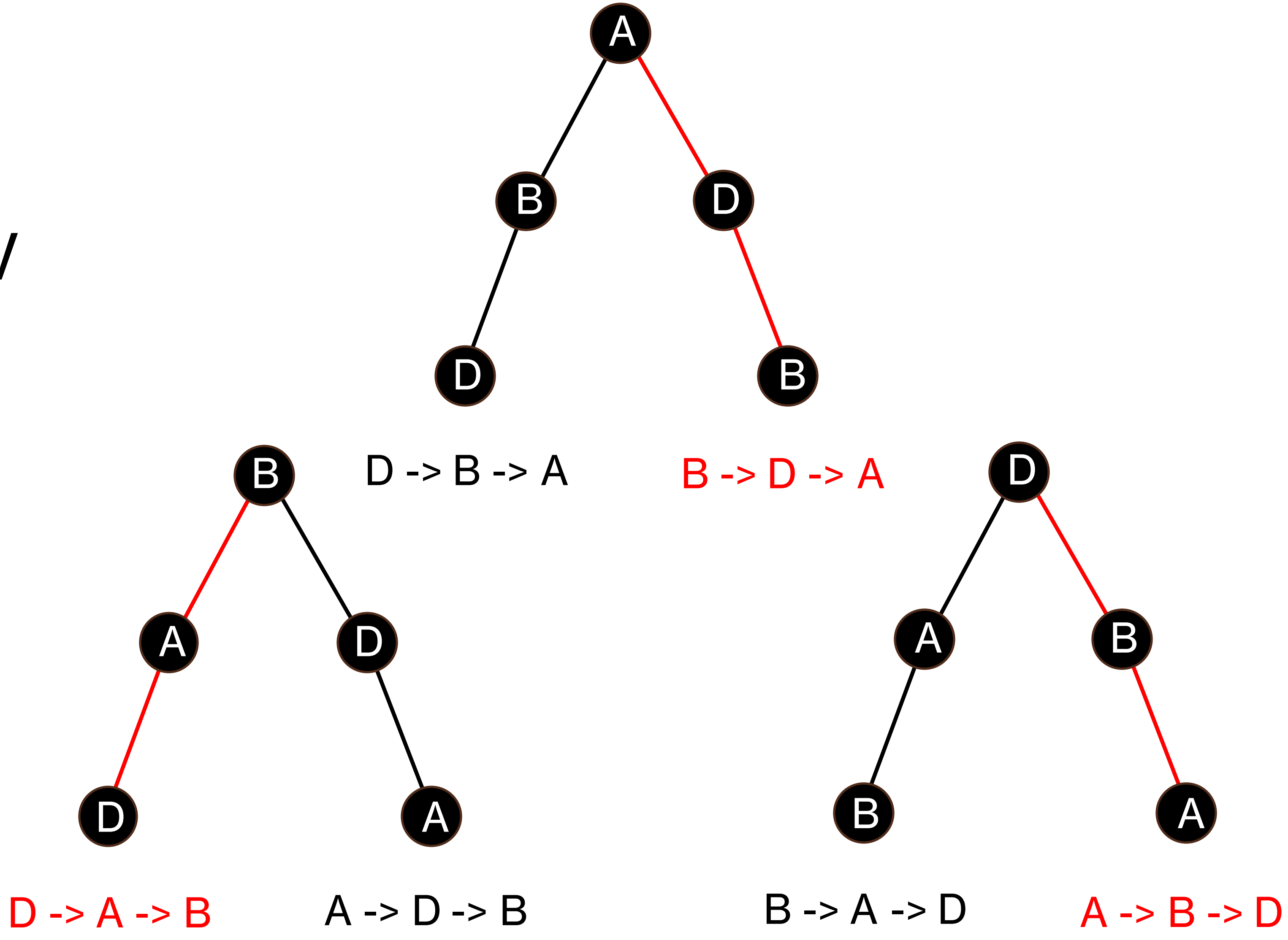
Visualizing IRV Outcomes

3 Candidate IRV
- Alice (A)
- Bob (B)
- Diego (D)

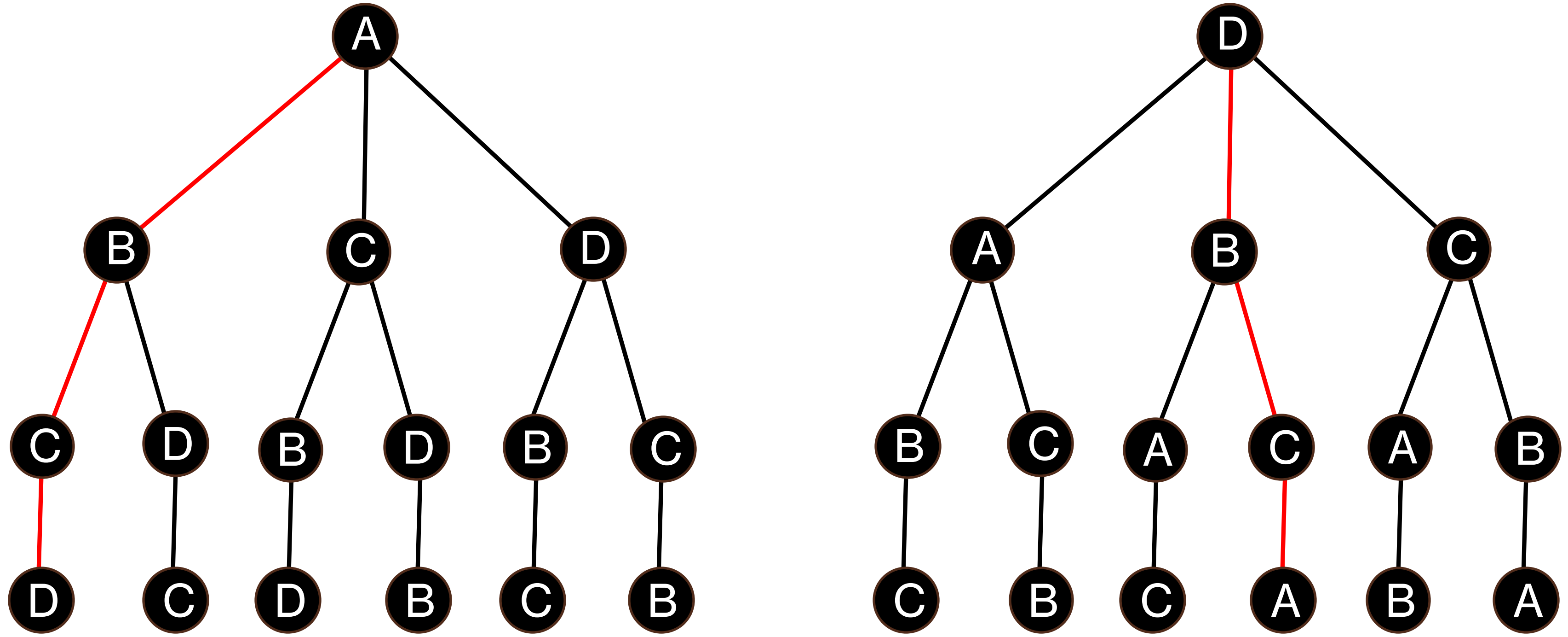


Visualizing IRV Outcomes

3 Candidate IRV
- Alice (A)
- Bob (B)
- Diego (D)



Visualizing IRV Outcomes



D->C->B->A D->B->C->A C->B->D->A
C->D->B->A B->D->C->A B->C->D->A

C->B->A->D C->A->B->D B->A->C->D
B->C->A->D A->C->B->D A->B->C->D

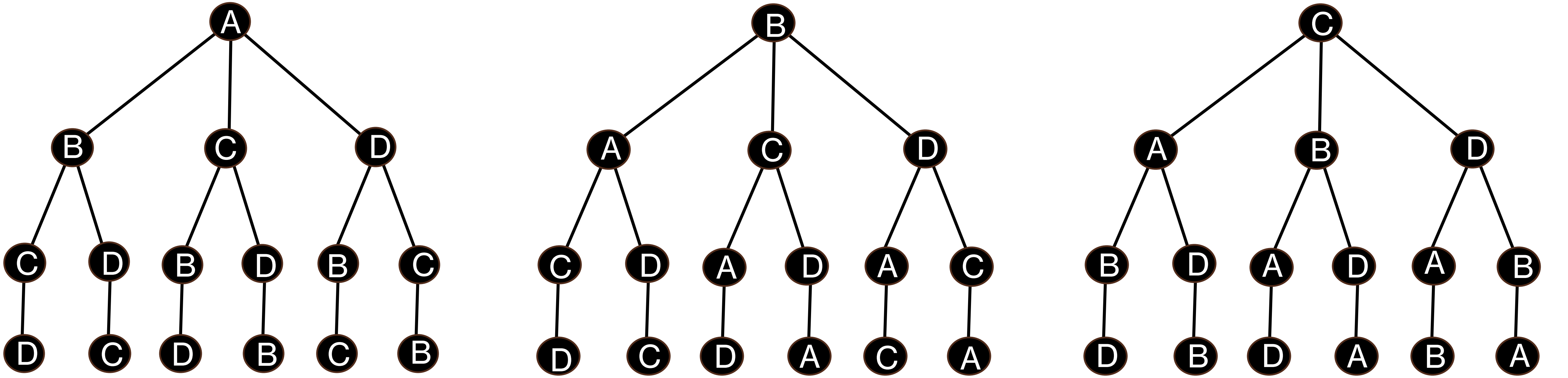
4 Candidate IRV
- Alice (A)
- Bob (B)
- Chuan (C)
- Diego (D)

+ Two more

(24 possible orders)

Visualizing (Alternate!) IRV Outcomes

Diego (D) → Reported Winner!

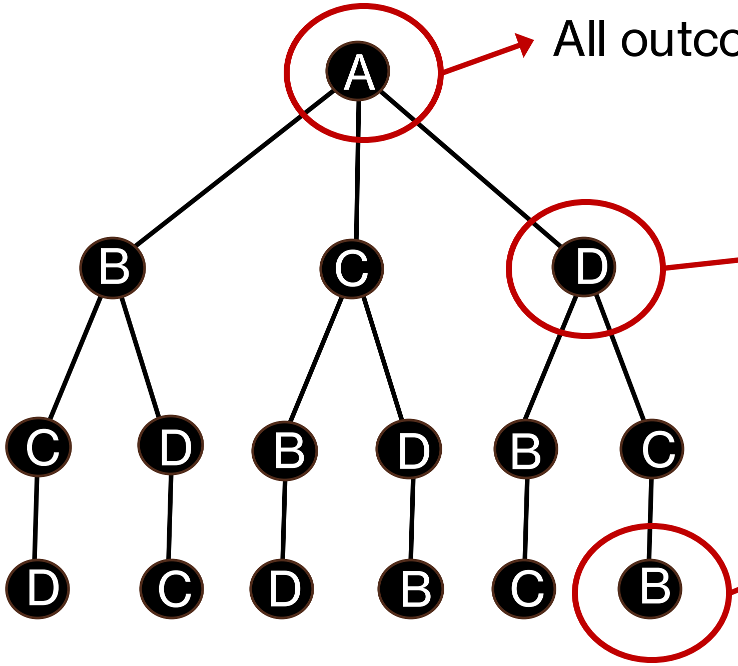


D->C->B->A D->B->C->A C->B->D->A
C->D->B->A B->D->C->A B->C->D->A

D->C->A->B D->A->C->B C->A->D->B
C->D->A->B A->D->C->B A->C->D->B

D->B->A->C D->A->B->C B->A->D->C
B->D->A->C A->D->B->C A->B->D->C

What does each node represent?



All outcomes that end with Alice (A) as the winner.

All outcomes that end with Alice (A) as the winner, and Diego (D) as the runner-up.

The specific outcome where Bob (B) is eliminated first, Chuan (C) next, Diego (D) is the runner-up and Alice (A) is the winner.

Our leaves are complete outcomes while each intermediate node describes a set of outcomes.

Exercise

Tally the following example IRV election.

Preference order	Number of ballots
(A, B, C, D)	50
(A, C)	40
(B, C, A)	25
(B, D, A)	25
(C, A, B)	30
(C, D, B)	45
(D)	100

Draw the elimination tree for the case where D is the winner.

Exercise (Solution)

Tally the following example IRV election.

Preference order	Number of ballots
(A, B, C, D)	50
(A, C)	40
(B, C, A)	25
(B, D, A)	25
(C, A, B)	30
(C, D, B)	45
(D)	100

Initial (first preference) tallies:
A : 90
B : 50
C : 75
D : 100

B is eliminated, giving 25 votes to C and 25 to D.
A : 90
C : 100
D : 125

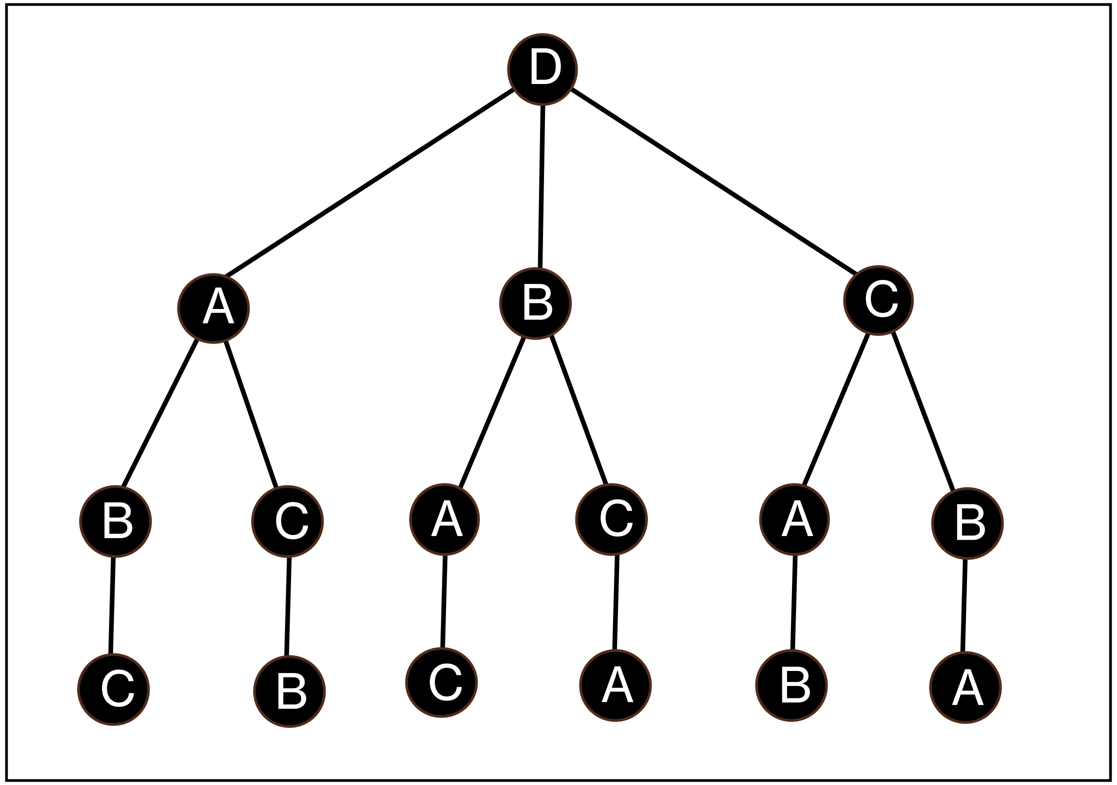
A is eliminated, giving 90 votes to C.
C : 190 (C wins!)
D : 125

Draw the elimination tree for the case where D is the winner.

Exercise (Solution)

Tally the following example IRV election.

Preferences	**Count**
(A, B, C, D)	50
(A, C)	40
(B, C, A)	25
(B, D, A)	25
(C, A, B)	30
(C, D, B)	45
(D)	100

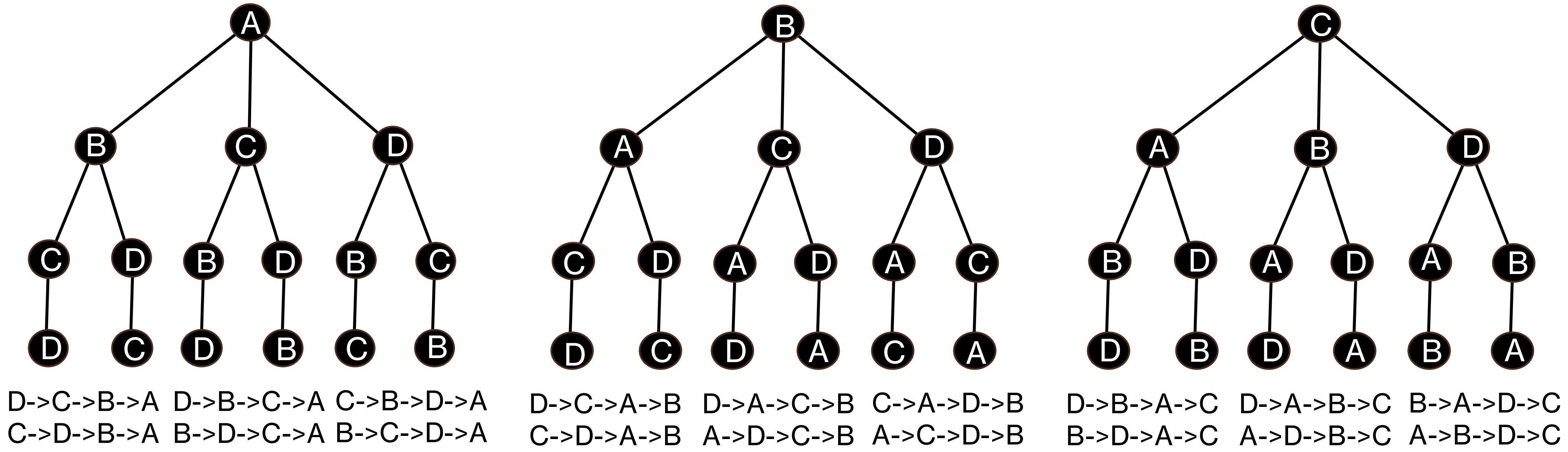


Draw the elimination tree for the case where D is the winner.

Assertions

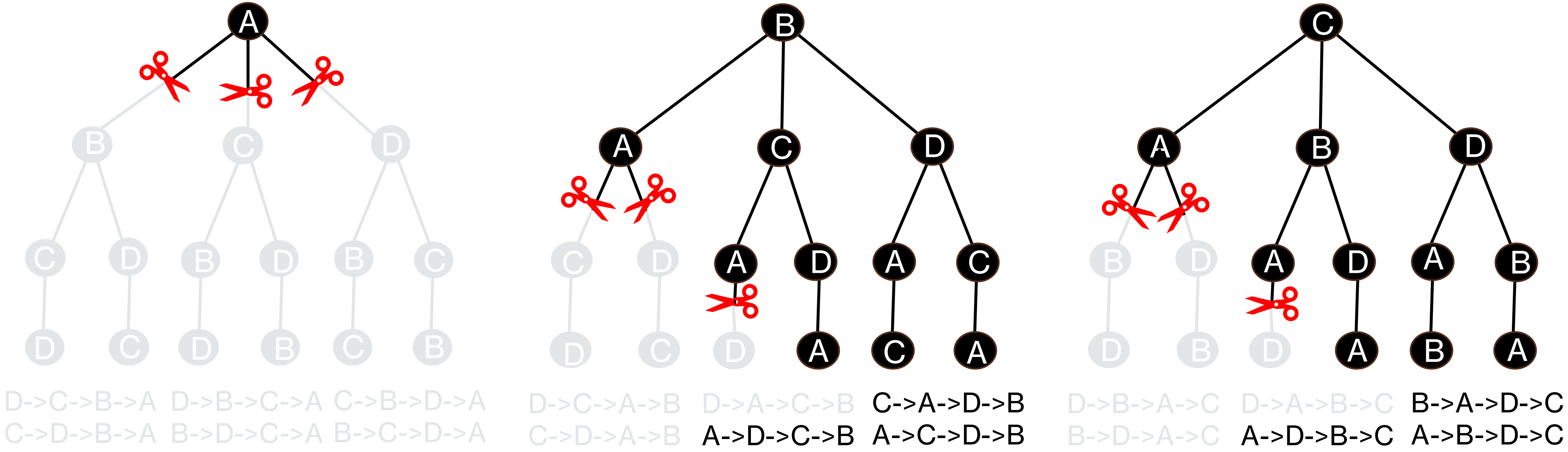
Ruling Out (Alternate!) IRV Outcomes

Assertion!



Ruling Out (Alternate!) IRV Outcomes

Diego cannot be eliminated before Alice



Assertion Types

Not Eliminated Before (NEB)
(Never Eliminated Before)

Not Eliminated Next (NEN)

Assertion Types ✂️ Not Eliminated Before (NEB) (Never Eliminated Before)

Diego NEB Alice

The *maximum* tally Alice could ever have is less than the *minimum* tally Diego will ever have.

So, Diego will *always* have more votes than Alice!

Preference order	Number of ballots
A C B	1000
D	3000
C A D	500
B A	1000
C D A	400

Minimum Tally for Diego: 3000 votes

Maximum Tally for Alice: 1000 + 500 + 1000 = 2500 votes

Assertion Types ✂ Not Eliminated Next (NEN)

NEN: Diego > Alice when only {Diego, Alice} remain

In the context where we assume everyone other than {Diego, Alice} have been eliminated, Diego has more votes than Alice.

Preference order	Number of ballots
A C B	1000
D	3000
C A D	500
B A	1000
C D A	400

Tally for Diego: 3000 + 400 = 3400 votes

Tally for Alice: 1000 + 500 + 1000 = 2500 votes

Exercise

Think about the following 3 assertions. Write out in your own words what they mean

Alice NEB Bob

NEN: Alice > Chuan if only {Alice, Chuan} remain

NEN: Alice > Chuan if only {Alice, Chuan, Diego} remain

Exercise

Consider an IRV election with four candidates: Alice, Bob, Diego, and Chuan. Suppose you are given a set of three assertions:

Alice NEB Bob

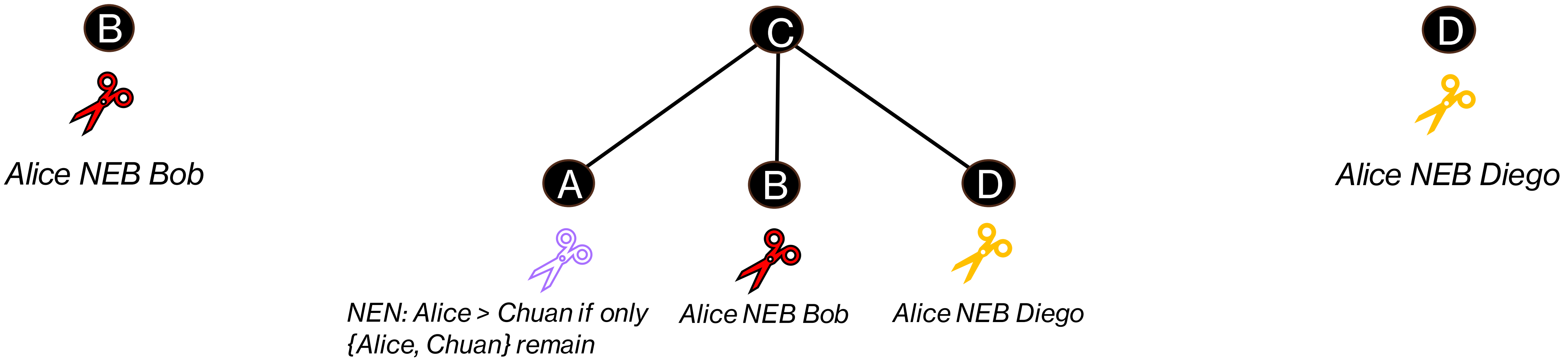
Alice NEB Diego

NEN: Alice > Chuan if only {Alice, Chuan} remain

Does this imply that Alice won? Either argue that it does, or provide an alternate winner via an elimination order that is consistent with these three assertions.

Exercise (Solution)

Alice NEB Bob   Alice NEB Diego   NEN: Alice > Chuan if only {Alice, Chuan} remain



Auditing Assertions

Scoring NEB Assertions

Example: Alice NEB Bob

This says that Alice's first preferences exceed the total number of mentions of Bob that are not Preceded by a higher preference for Alice.

Fits into existing RLA, but with our two candidates being "Alice 1st Preference" and "Bob mention"

Ballot contents	**Counted for**	**Example**
First preference for Alice	Alice 1st Preference	(A, B, C, D)
Bob mention, *no* higher preference for Alice	Bob mention	(C, B, D, A)
Bob mention, *with* higher preference for Alice	Neither	(C, A, B)
Anything else	Neither	(C, A, D)

Auditing NEB Assertions

Example: Alice NEB Bob

We randomly sample ballots, compare what is on the paper to its matching CVR, and determine whether there are discrepancies.

Overstatement: Error that mistakenly records a first preference for Alice or omits a mention of Bob not preceded by Alice.

One vote overstatement: CVR showing (A, C) and ballot paper (D, C)

Two vote overstatement: CVR showing (A, B, C) and ballot paper (C, B, A)

Scoring NEN Assertions

Example: NEN: Alice > Bob if only {Alice, Bob, Chuan} remain

This says that Alice has more votes than Bob when only Alice, Bob, and Chuan are continuing.

Fits into existing RLA, but with our two candidates being “Alice’s tally when Alice, Bob, and Chuan remain” and “Bob’s tally when Alice, Bob, and Chuan remain”.

Ballot contents	Counted for	Example
Alice, not preceded by Bob or Chuan	Alice	(A, B, C, D)
Bob, not preceded by Alice or Chuan	Bob	(D, B, C, A)
Chuan, not preceded by Alice or Bob	Neither	(D, C)
Anything else	Neither	(D)

Auditing NEN Assertions

Example: NEN: Alice > Bob if only {Alice, Bob, Chuan} remain

We randomly sample ballots, compare what is on the paper to its matching CVR, and determine whether there are discrepancies.

Overstatement: An error that advantages Alice by mistakenly listing her as the highest preference among Alice, Bob, and Chuan, or disadvantages Bob by mistakenly not listing him as the highest preference among Alice, Bob, and Chuan.

One vote overstatement: CVR showing (A, C, D, B) and ballot paper (D, C, A, B)

Two vote overstatement: CVR showing (A, B, C) and ballot paper (D, B, C)

Exercise

Suppose we have a CVR (C, D, B, A) and the corresponding ballot says (D, C, B, A).

Is this a one or two vote overstatement (or neither) for the following assertions?

Assertion	**Overstatement**
Chuan NEB Alice	
Chuan NEB Diego	
NEN: Chuan > Bob if only {Alice, Bob, Chuan} remain	

Exercise (Solution)

Suppose we have a CVR (C, D, B, A) and the corresponding ballot says (D, C, B, A).

Is this a one or two vote overstatement (or neither) for the following assertions?

Assertion	**Overstatement**
Chuan NEB Alice	1
Chuan NEB Diego	2
NEN: Chuan > Bob if only {Alice, Bob, Chuan} remain	0

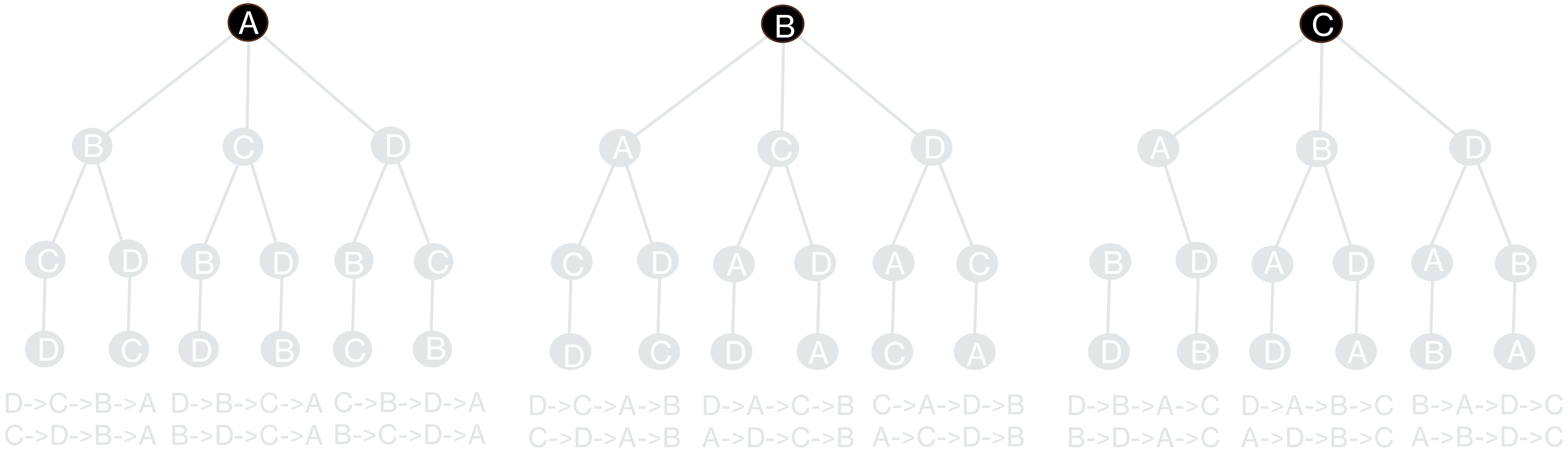
How RAIRE Generates Assertions

Objectives

- We need to find an assertion to rule out every branch in our collection of alternate outcome trees.
- We **do not** want to create or explore these trees in their entirety!
- We want to minimize the number of ballots auditors will have to collect.

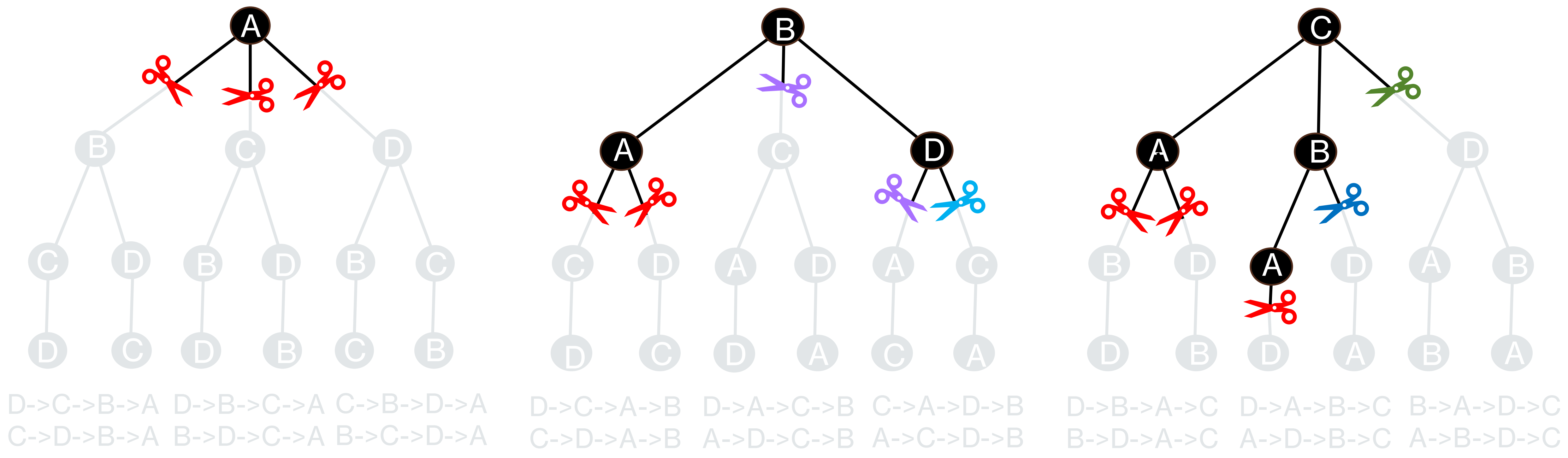
Simple (but Sub-Optimal) Approach

What if we just start with the top level of our alternate outcome trees and continue exploring down each branch *until* we know how to rule it out with an assertion?



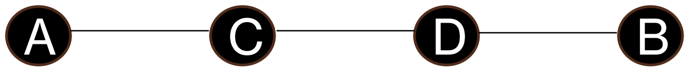
Simple (but Sub-Optimal) Approach

This would work, and give us a valid set of assertions, but they might be expensive!

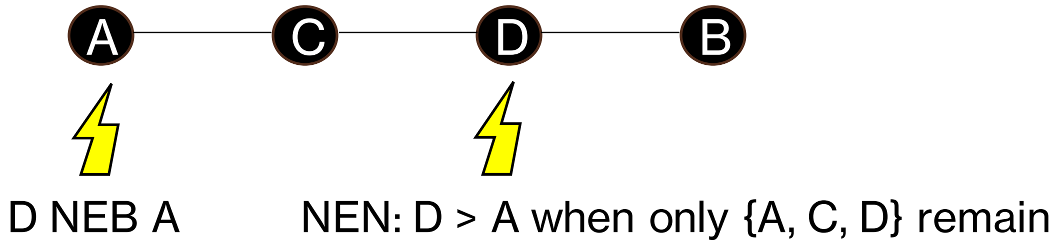


RAIRE's Approach

- Consider one branch in our set of alternate outcomes trees.



- There may be multiple points at which we could attack outcome.



- A branch's **weakest point** is the point at which it can be attacked with the cheapest to audit assertion.

RAIRE’s Approach

1. Find the branch whose weakest point requires the most expensive assertion to audit.

2. The cost of this assertion gives us a lower bound on the overall cost of our audit.

3. Find assertions <u>with costs within this bound</u> to rule out all other branches, exploring only enough of each branch until a weak enough point has been found.

A Guide to Risk Limiting Audits for Instant Runoff Voting (IRV)

Vanessa Teague[1] Michelle Blom[1,2] Peter Stuckey[3]
Andrew Conway Mark Pinoli[1]

[1] Democracy Developers Ltd.
[2] School of Computing and Information Systems, The University of Melbourne
[3] Faculty of IT, Monash University

Guide (Parts 1 and 2):
https://github.com/DemocracyDevelopers/Colorado-irv-rla-educational-materials

An online assertion visualizer and explainer: https://democracydevelopers.github.io/raire-rs

© Democracy Developers Ltd.