# **Lean pre-coding sheet — 71/89**

**Component:** fixtures/annex\_b/part\_0/ballots.json (Part 0 fixture: **BallotTally** dataset)  
 **Version/FormulaID:** Data fixture (not part of FID; FID covers rule primitives only).

## **1) Goal & success**

**Goal:** Provide the canonical **BallotTally** for Part 0 in the exact shape required for each ballot type. Must align with Registry options/order and with ParameterSet variables.

**Success:** Schema-valid; **tally sanity** holds per unit; deterministic option order respected; loads into pipeline and drives **TABULATE** correctly.

## **2) Scope**

**In scope:** One **BallotTally** dataset with ID/label; per-unit tallies **by ballot type** (approval/plurality/score/ranked IRV/ranked Condorcet).

**Out of scope:** Parameter values (separate fixture), allocation/aggregation logic, reporting prose.

## **3) Inputs → outputs**

**Input artifact:** fixtures/annex\_b/part\_0/ballots.json (BallotTally).

**Used by pipeline:** Feeds **TABULATE** (step 2) after **VALIDATE**; then flows into allocation/aggregation.

## **4) Entities/Tables (minimal)**

## **5) Variables (only ones used here)**

## **6) Functions (signatures only)**

N/A (fixture only).

## **7) Algorithm outline (how it’s consumed)**

**VALIDATE** checks **tally sanity** per unit: Σ(valid option tallies) + invalid\_or\_blank ≤ ballots\_cast.

**TABULATE** interprets shape per **VM-VAR-001**: plurality→vote counts; approval→approval counts; score→score sums (with scale/normalization context); IRV→round logs with exhaustion; Condorcet→pairwise from rankings.

**Gate denominators:** approval gate is **fixed** to **approval rate = approvals\_for\_change / valid\_ballots** (not approvals share). Others use support/valid ballots unless **VM-VAR-007=on** (valid+blank for **gates only**).

## **8) Fixture shapes (must match exactly) — per Annex B Part 0**

**Approval:** per Unit: ballots\_cast, invalid\_or\_blank, approvals { Option → count }.

**Plurality:** per Unit: ballots\_cast, invalid\_or\_blank, votes { Option → count }.

**Score:** per Unit: ballots\_cast, invalid\_or\_blank, score\_sum { Option → sum }, ballots\_counted; plus **scale** (VM-VAR-002..003) and **normalization** (VM-VAR-004).

**Ranked IRV:** rounds[{ ranking[], count }]; **exhaustion policy** is reduce\_continuing\_denominator (VM-VAR-006).

**Ranked Condorcet:** ballots[{ ranking[], count }]; completion rule per **VM-VAR-005**.

## **9) State flow (very short)**

Loaded at **LOAD**; validated at **VALIDATE**; consumed by **TABULATE** to produce UnitScores, which then feed allocation/aggregation.

## **10) Determinism & numeric rules**

**Stable option order** (by Option.order\_index) and **sorted JSON keys**; counts are integers; presentation rounding occurs only in reports (one decimal).

**Approval gate denominator** remains **approval rate**; internal comparisons use **round half to even**.

## **11) Edge cases & failure policy**

**Mismatch with registry options/order**; **negative counts**; **sum tallies > ballots\_cast**; missing scale/normalization for score; malformed IRV rounds or Condorcet rankings → **VALIDATE error**; run goes down **Invalid** path.

**WTA interplay:** if later allocation\_method=winner\_take\_all, ensure involved units have magnitude=1 (checked elsewhere but affects acceptance).

## **12) Test checklist (must pass)**

Schema validates for the selected **ballot\_type**; **tally sanity** passes in all units.

Baseline **6A** cases reproduce expected allocations (PR 1–2–3–4; WTA winner D; LR/D’Hondt/Sainte-Laguë → 3–2–2).