Visualizing QCA

Claude Rubinson
University of Houston—Downtown

4th International QCA Experts Workshop
University of Zürich
7 December 2016

Objects in a QCA Analysis

- Calibrated data sets
- Truth tables
- Consistency/coverage solutions

Goals of QCA Visualization

- Present superset/subset relationships
- Preserve case holism & diversity
- Clarify configurations
- Convey the range of solution complexity

Examples

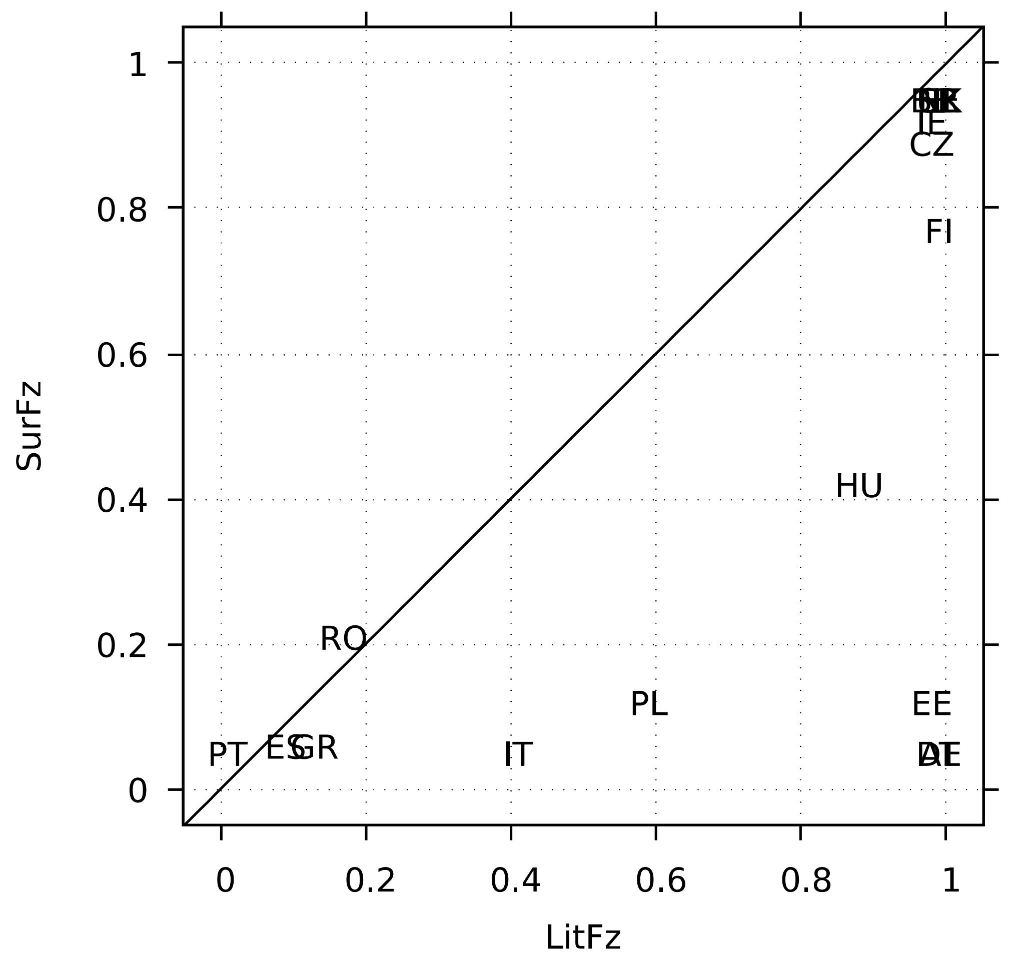
- Rihoux & Ragin (2008) *Config Comp Methods*
- Ragin & Fiss (2008) *Redesigning Social Inquiry*

Calibrated Data: 2x2 Tables

	National Literacy Rate (LitCr)	
	Not High	High
Democracy Survival	—	BE, CZ, FI, FR, IE, NL, SE, UK
Democracy Breakdown	ES, GR, IT, PT, RO	AT, DE, EE, HU, PL

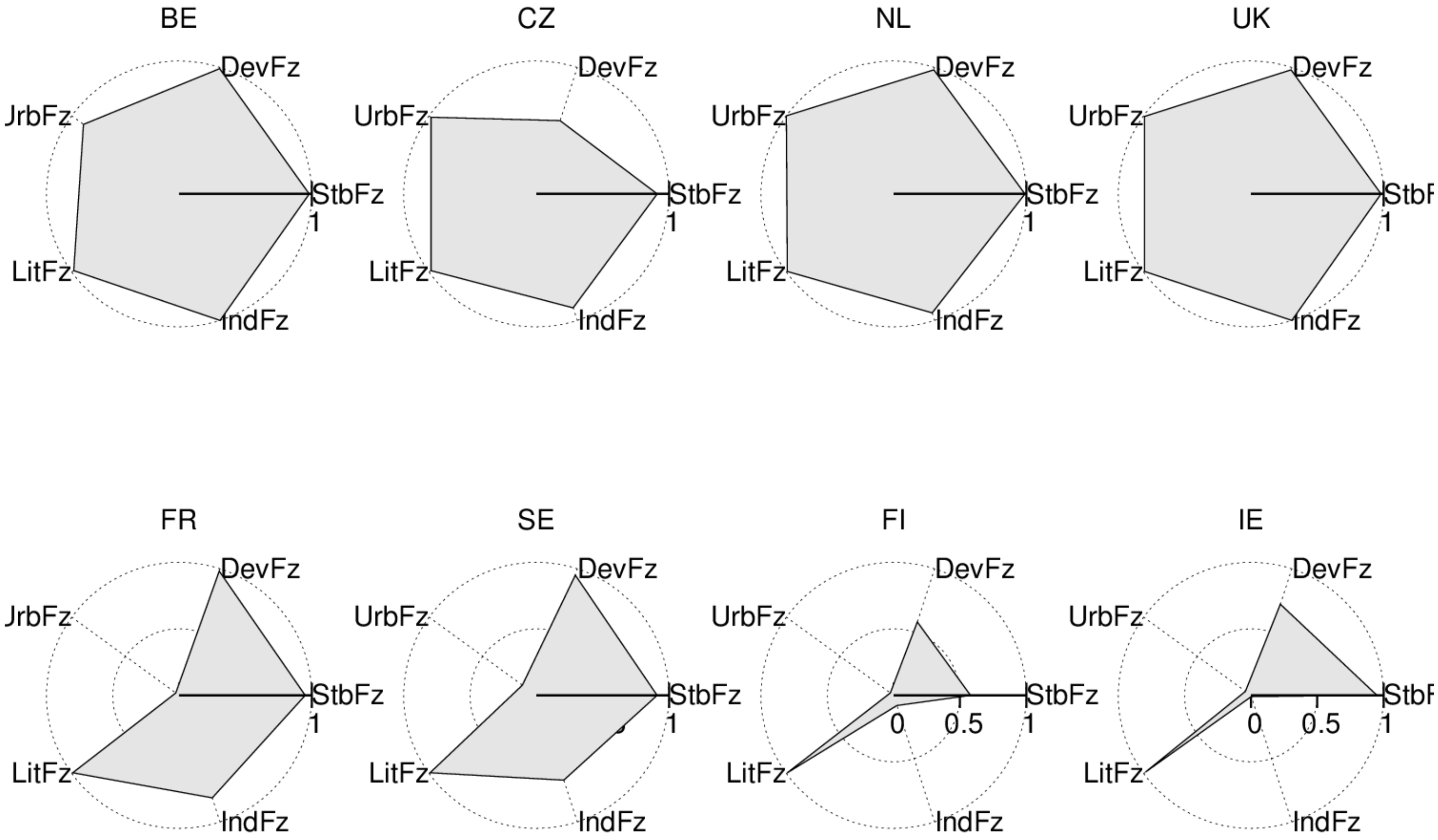
- For crisp sets
- Easy to construct
- Easy to interpret, but need to explain necessity/sufficiency

Calibrated Data: Scatterplots



- For fuzzy sets
- Square aspect ratio
- Diagonal reaches frame corners and is same weight
- Easy to construct
- Easy to interpret, but need to explain triangular plots and necessity/sufficiency

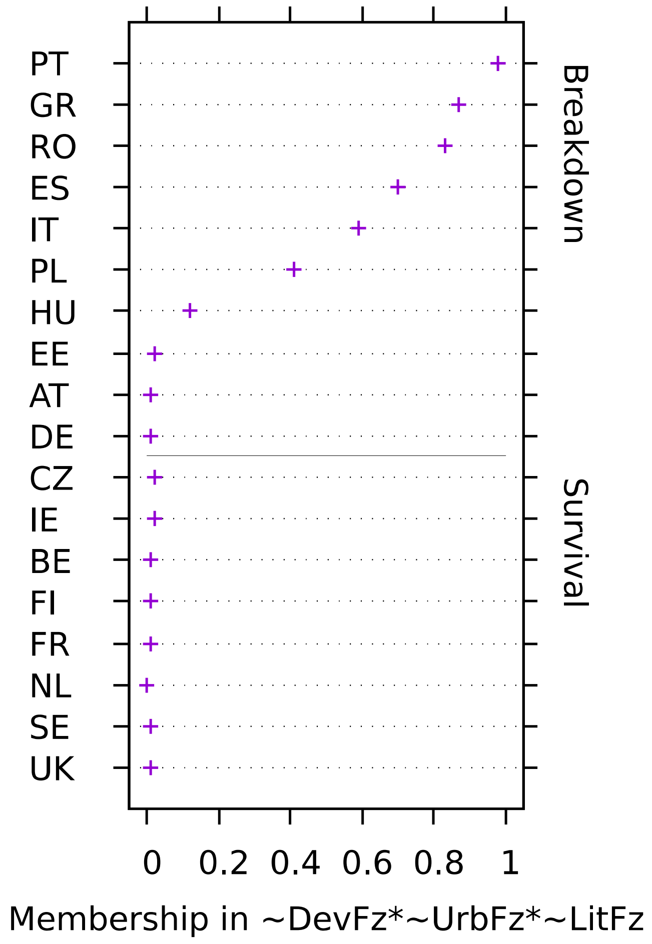
Calibrated Data: Radar Charts



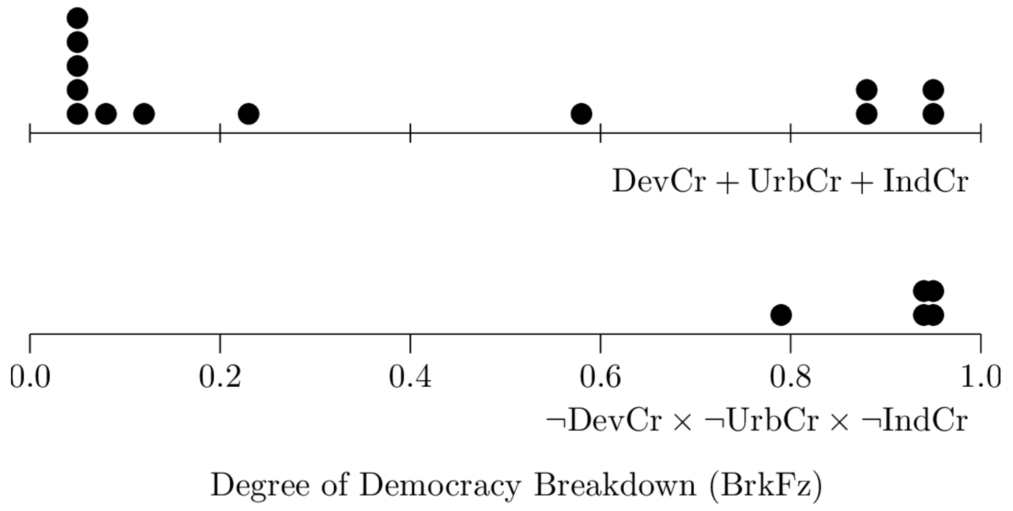
- Compare *shape* of observations, using fuzzy sets
- Compare configurations by aggregating (e.g., min, mean, max) across observations (Meuer, et. al. 2015)

Calibrated Data

Fuzzy set crossed with crisp set



Rank-order plot
(a.k.a., Cleveland dot plot)

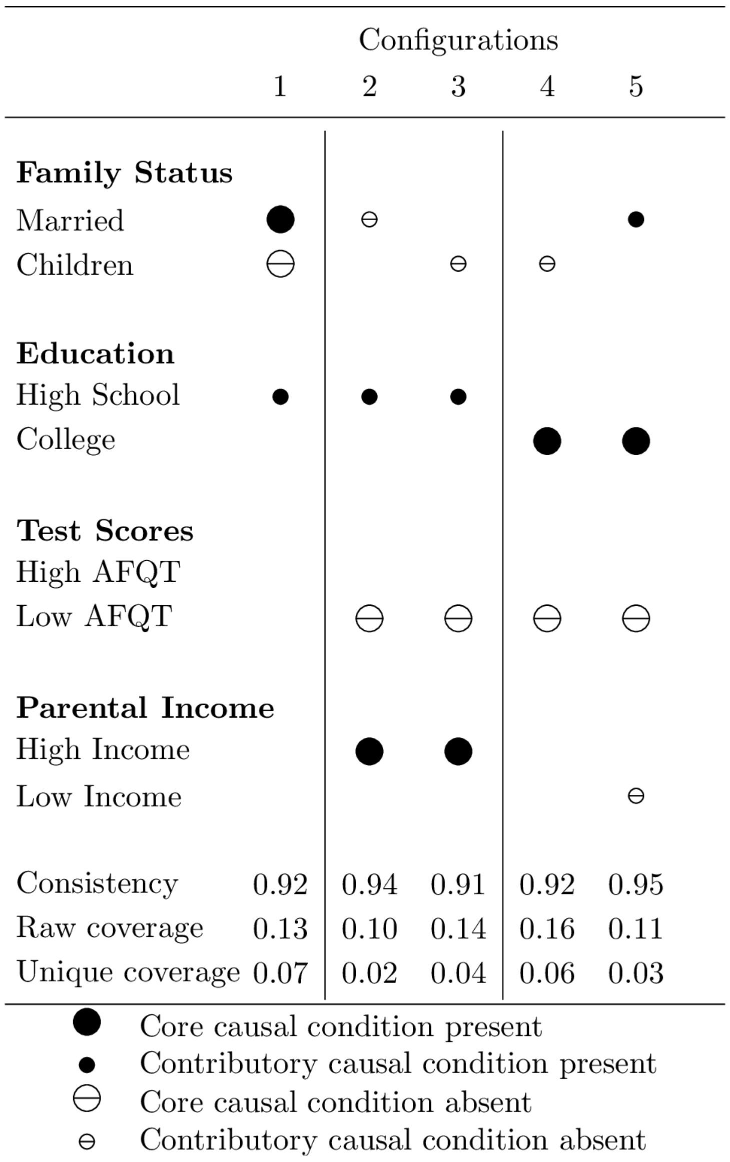


Dot plot

Consistency/Coverage Solutions

Fiss configuration charts

- Displays all configurations and how they relate
- Simultaneously present parsimonious and intermediate (or complex) solutions
- Order of configurations is up to researcher; grouping by core conditions is just one option
- Instead of numbering configurations, use meaningful names

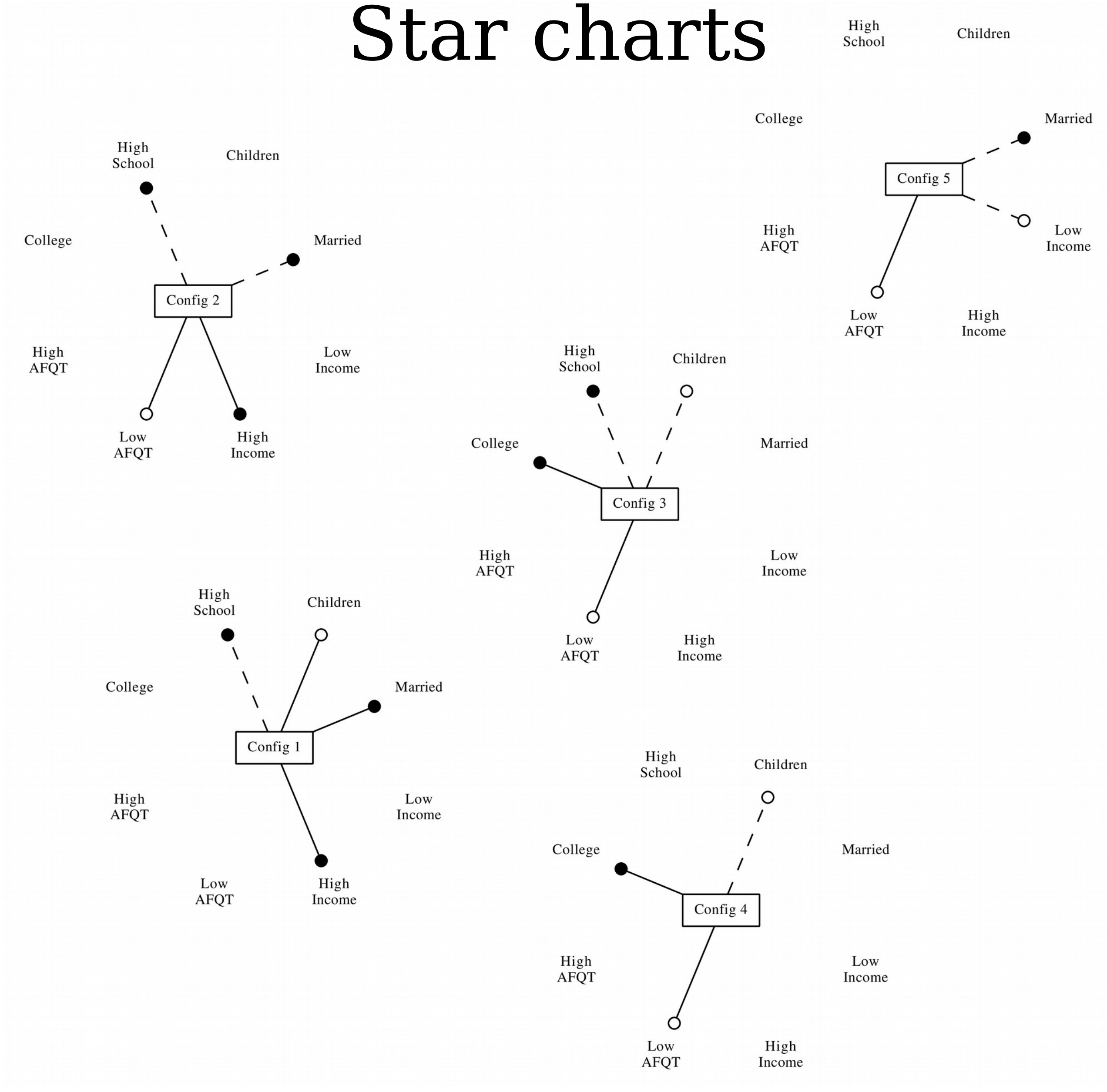


	Configurations				
	1	2	3	4	5
Family Status					
Married	●	⊝			•
Children	⊖		⊝	⊝	
Education					
High School	•	•	•		
College				●	●
Test Scores					
High AFQT					
Low AFQT		⊖	⊖	⊖	⊖
Parental Income					
High Income		●	●		
Low Income					⊝
Consistency	0.92	0.94	0.91	0.92	0.95
Raw coverage	0.13	0.10	0.14	0.16	0.11
Unique coverage	0.07	0.02	0.04	0.06	0.03

●	Core causal condition present
•	Contributory causal condition present
⊖	Core causal condition absent
⊝	Contributory causal condition absent

Consistency/Coverage Solutions

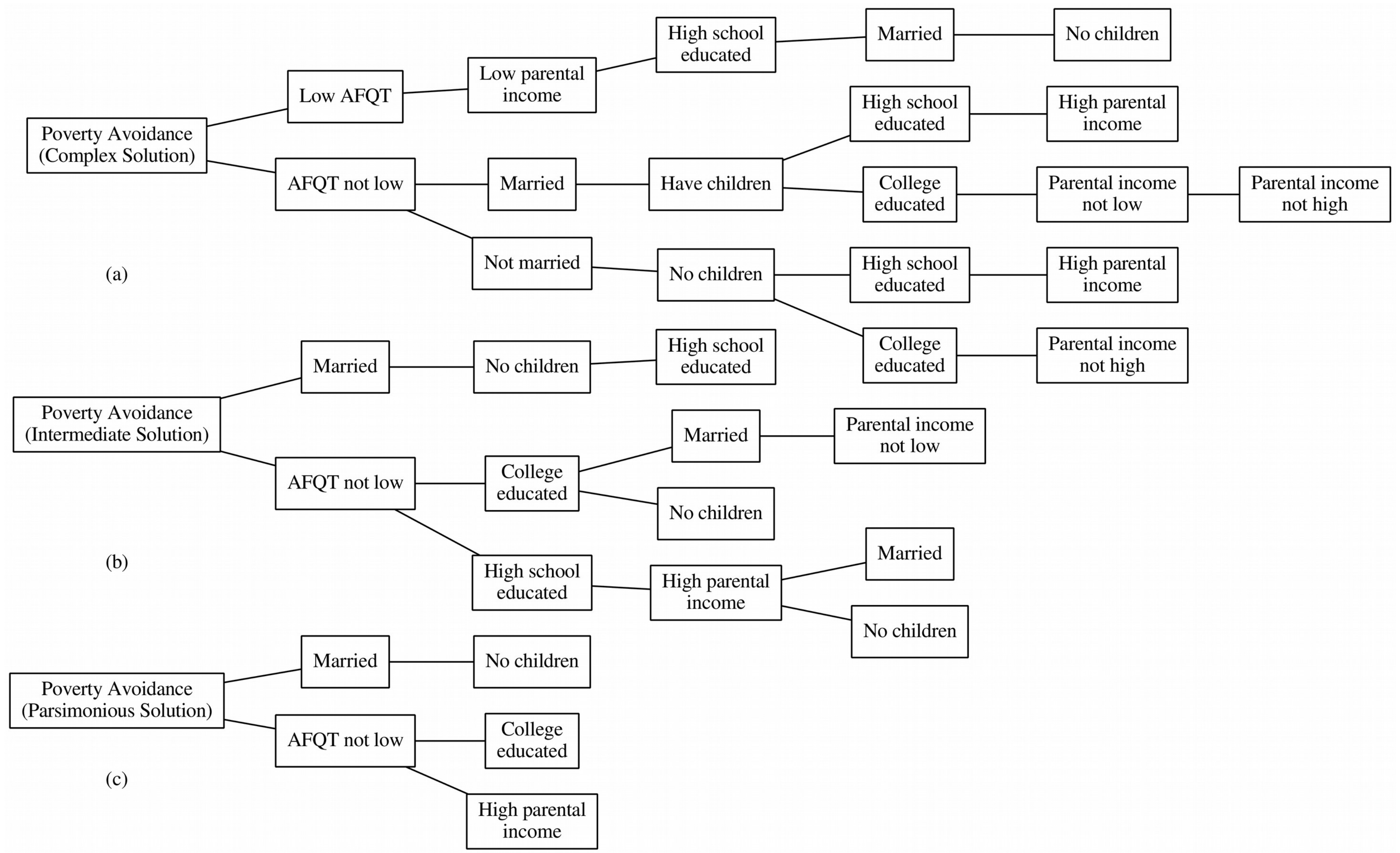
Star charts



● Condition present — Condition is part of parsimonious solution

○ Condition absent - - - Condition is part of intermediate solution

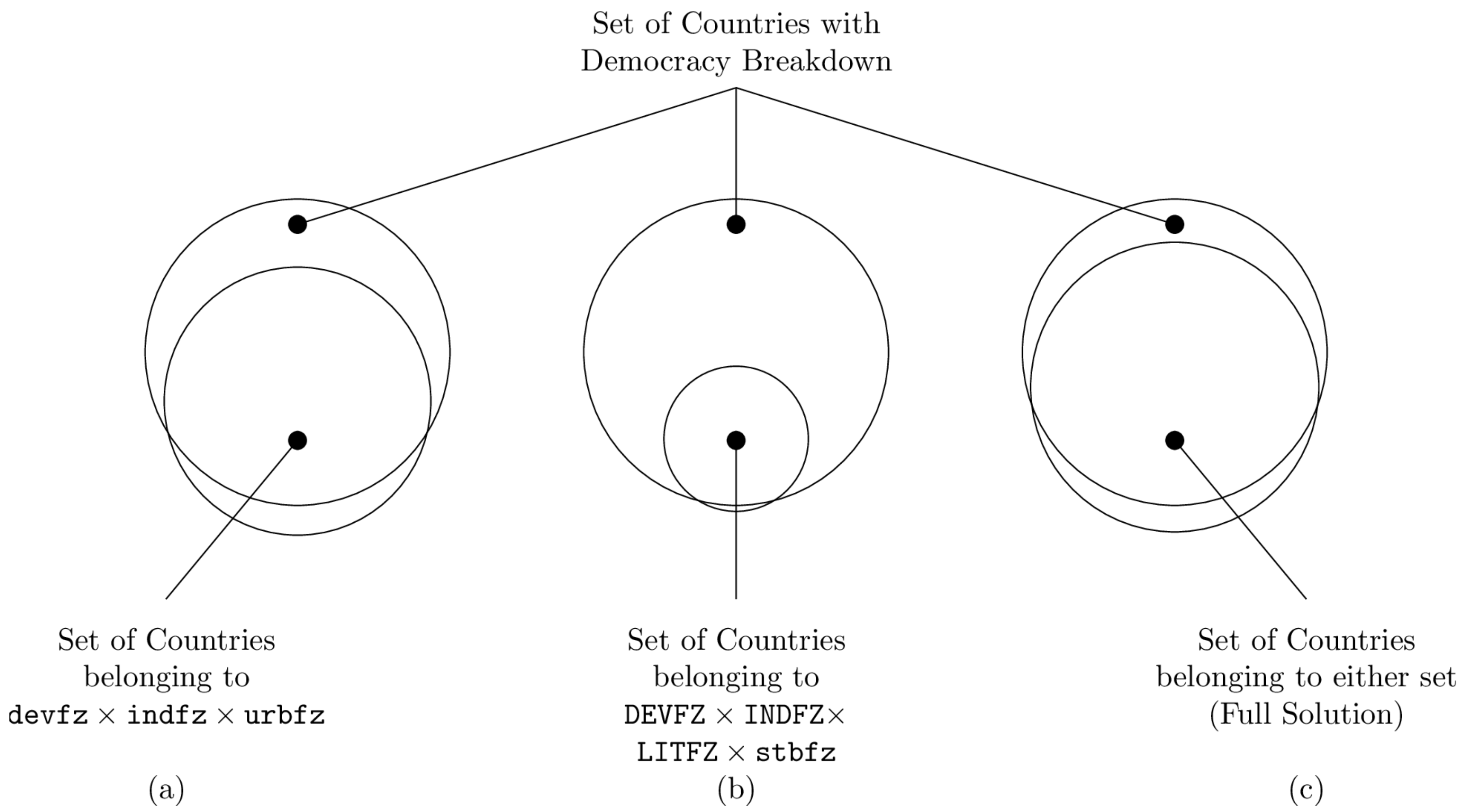
Consistency/Coverage Solutions
Branching diagrams (dendrograms)



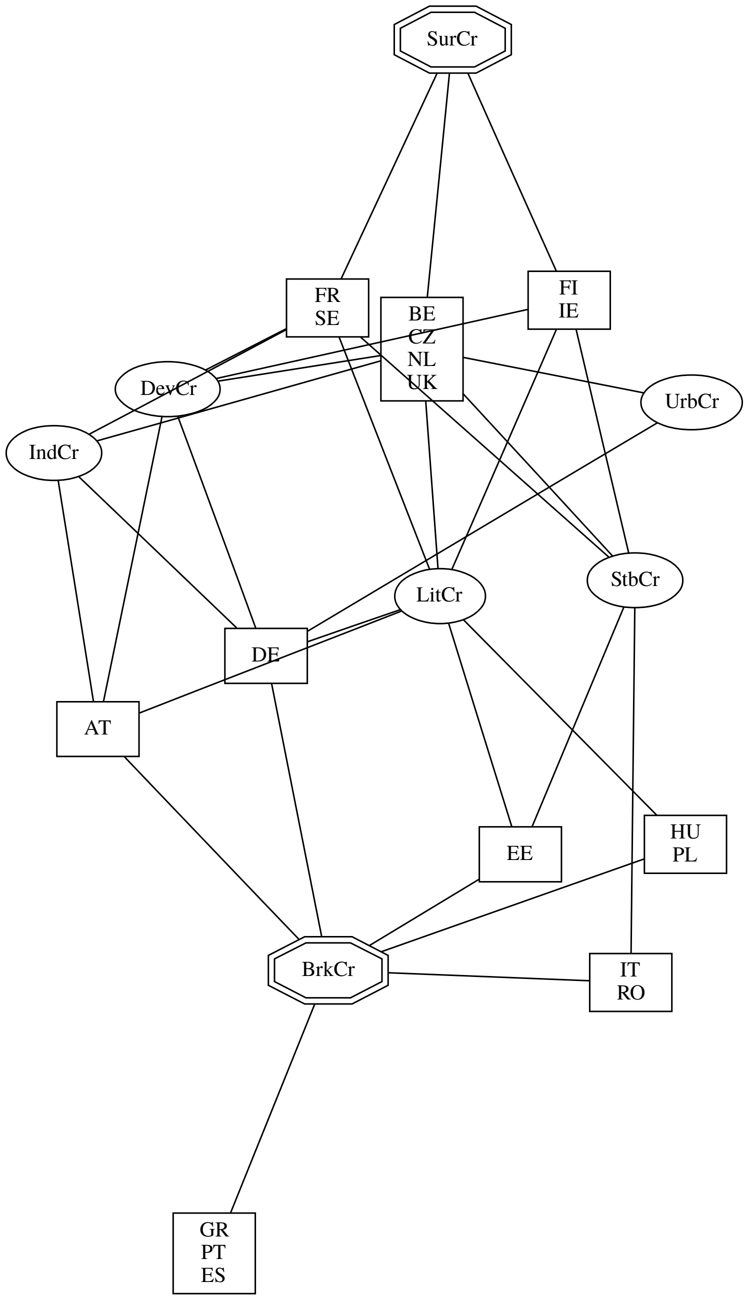
Superset/Subset Relationships

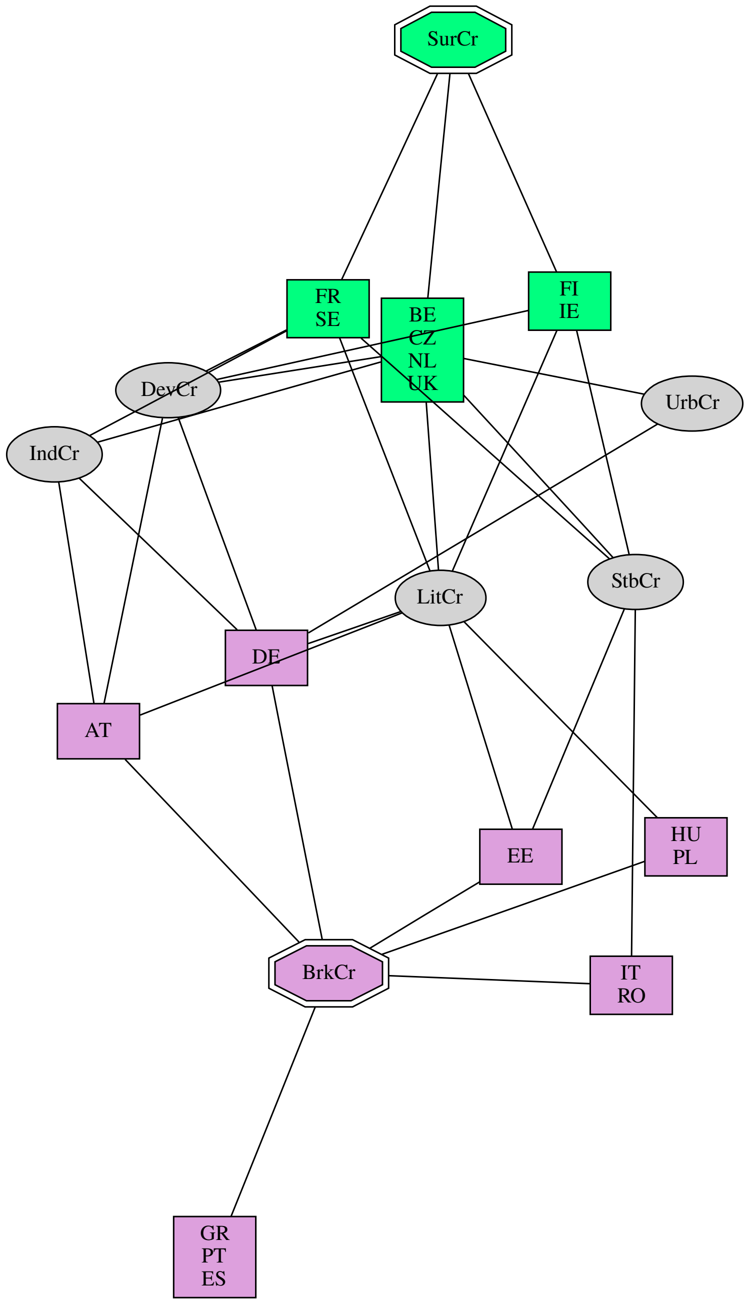
- Venn/Euler diagrams
  - Familiar and easy to interpret, but:
    - Low information density
    - Interpretability decreases as intersections increase
    - Difficult to convey proportionality
    - Programmatically generating area-proportional Euler diagrams with more than 3 sets is an unsolved problem
- Alternatives:
  - Force-directed graphs
  - Galois lattices
  - Linear diagrams

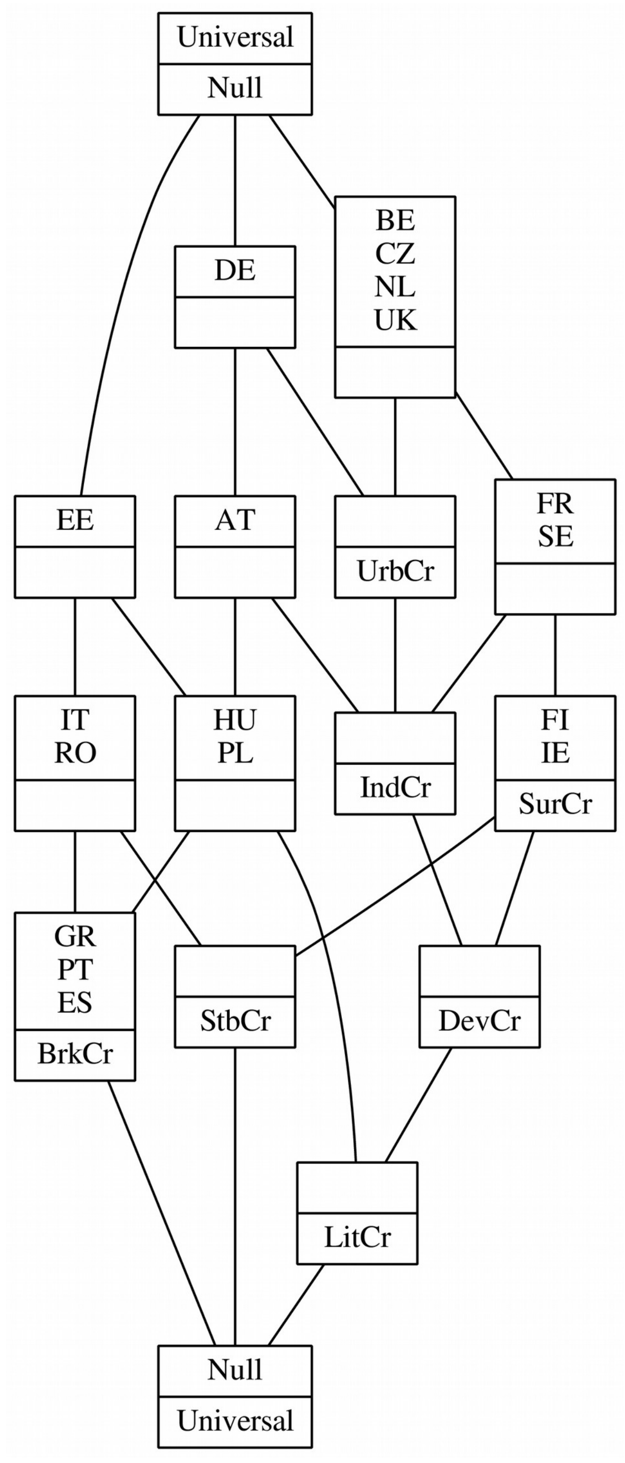
Area-Proportional 2-Set Venns


Set of Countries with Democracy Breakdown
Set of Countries belonging to $\texttt{devfz} \times \texttt{indfz} \times \texttt{urbfz}$
Set of Countries belonging to $\texttt{DEVFZ} \times \texttt{INDFZ} \times \texttt{LITFZ} \times \texttt{stbfz}$
Set of Countries belonging to either set (Full Solution)
(a)
(b)
(c)

Force-directed Graphs



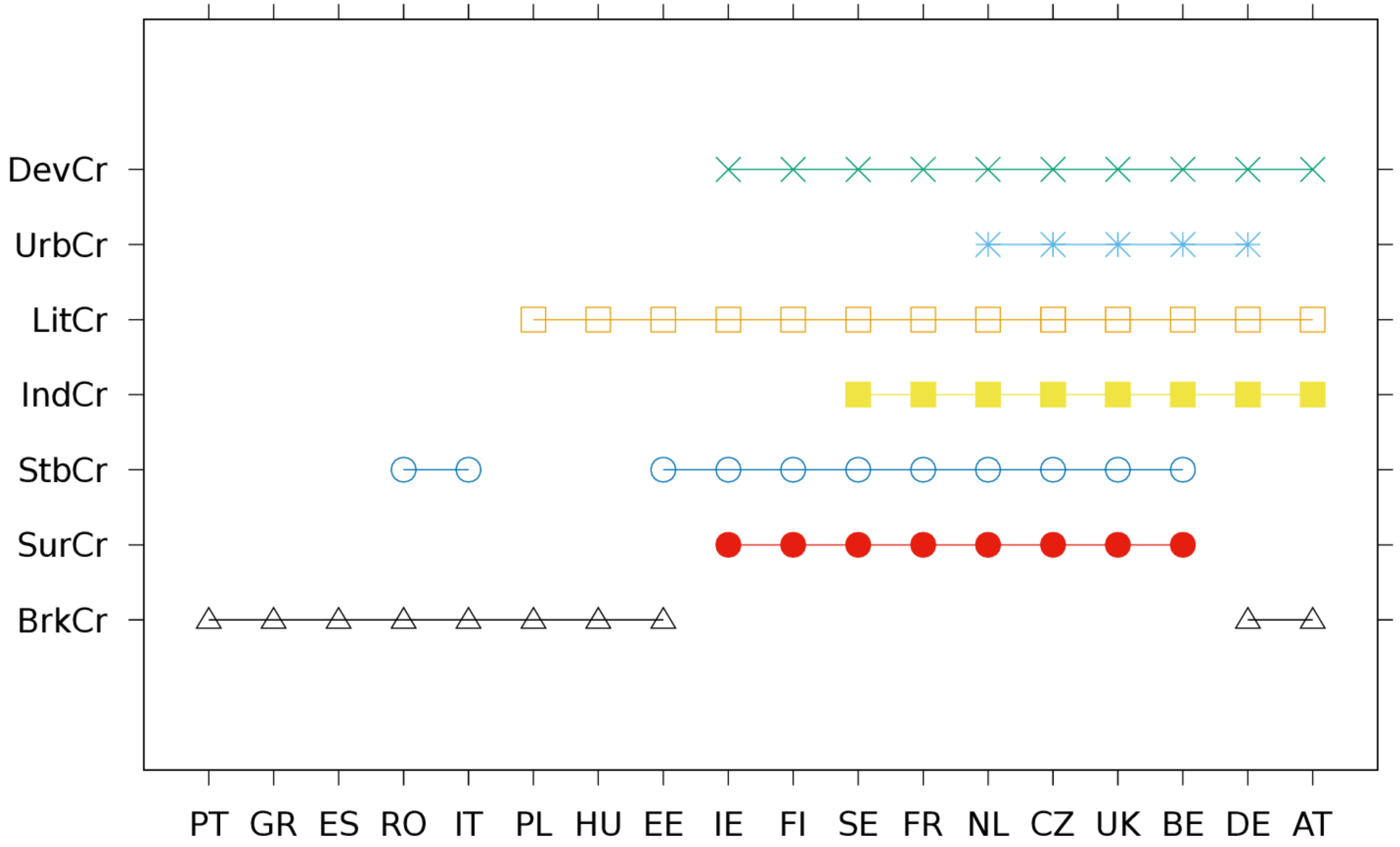




Galois Lattices

- Easy to construct using software (but not by hand)
- Not intuitive; can be difficult to interpret. Will need to interpret for reader.
- Presents superset/subset relationships simultaneously
- Requires crisp sets
- Particularly well-suited for depicting truth tables (QCAViz can include remainders)

Linear Diagrams



Software

- Visualizations presented here were initially produced using a variety of software (primarily Inkscape, GnuPlot, GraphViz, or Tik*Z)*
- Input data (calibrated data, truth tables, consistency/coverage solutions) typically require some processing to be visualized
- Variation in what can be automated, and to what extent; manual work always needed for best results
- Front-end scripts were written in various languages (typically awk, Bash, or Python)

QCAViz

Goals

- Focus on (small/medium-N) QCA
- Standardize inputs; automatically convert between objects
- Invoke backends as needed; invisible to user
- Relatively easy to add/update visualizations
- GUI for interactive use
- CLI for scripting

Workflow

Input
(Calibrated data, truth table, or consistency/coverage solution)

↓

Pre-processing

↓

Generate “backend” code
(GnuPlot, GraphViz, TikZ, etc)

↓

Post-processing

↓

Output:

- Render image, or
- Convert and save to SVG, EPS, etc., or
- Output raw code for producing image