AustMS 2020 : Think globally, act locally (20-25 minutes w/ slides & terminal)
09.12.20

2. Study automorphism groups of graphs / trees.
Infinite groups better ⇝ infinite graphs (fading)
Permutation topology
Stabilizer – local action on spheres – relationship to global structure
Study (finite) local actions in GAP

3. There are good theoretical & practical reasons.
4. "Don't take this slide too seriously"
In Lie theory, think unipotent matrices & kernel of adjoint rep.
For $\mathrm{Aut}(T_d)$, subgroup preserving natural bipartition (index 2) & $\{id\}$
Similar behavior for groups with appropriate local actions
Relationship between local and global properties.
5. Start with tree and label it such that ...
For a given k, take another, finite tree $B_{d,k}$, which is isomorphic to a k-ball in T_d. Then consider the unique colour-preserving homomorphism from $B_{d,k}$ to $B(x,k)$ (which sends b to x). ...
$\sigma_k(g,x)$ is the "k-local action of g at x"
6. closed not too hard to prove: if $g \notin U_k(F)$ then...
vertex-transitive: can make colour-preserving automorphisms
Local action is at most as big as F, could be less. It's an extension/compatibility problem.
7. To best describe the condition (C), think of an element of $\mathrm{Aut}(B_{d,k})$ as a collection of (k-1)-local actions.
"For every direction and for every element in F, that element can be extended in that direction."
8. All about subgroups of $\mathrm{Aut}(B_{d,k})$. Replace colours w/ numbers. Order leaves lexicographically w.r.t. numbering of paths.
Extensive manual – let me know if this or something similar could be useful to you.
9. Ph.D. position available.

```
F := AutB(3, 2);

a := Random(F);

LocalAction(1, 3, 2, a, []);

LocalAction(1, 3, 2, a, [1]);

b := CompatibleElement(3, 2, F, a, 1);

LocalAction(1, 3, 2, b, []);

LocalAction(1, 3, 2, b, [1]);


c := Random(AutB(3,5));

LocalAction(3, 3, 5, c, [2, 1]);


list := ConjugacyClassRepsCompatibleSubgroups(3, 2, F);
for H in list do Print(ImageOfProjection(3, 2, H, 1), "\n"); od;
———— " ————   StructureDescription(...)

ConjugacyClassRepsCompatibleSubgroupsWithProjection(3, 2, 1, SymmetricGroup(3));
list_s3 := last;
for H in list_s3 do Print(IsDiscrete(3, 2, H), "\n"); od;
```

?UGALY ⇝ explain