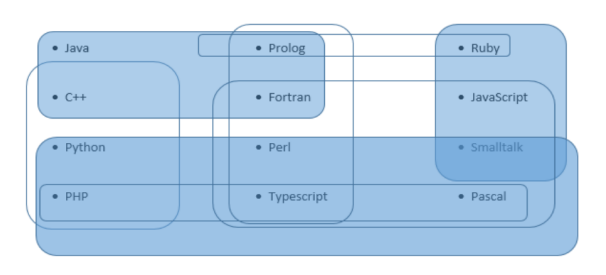
**ANALISIS COMPARATIVO ENTRE ALGORTIMO APROXIMADO Y BACKTRACKING**



**Cantidad de empleados:** 8

**Cantidad de lenguajes:** 12

* const string L1 = "Java";
* const string L2 = "C++";
* const string L3 = "Python";
* const string L4 = "PHP";
* const string L5 = "Prolog";
* const string L6 = "Fortran";
* const string L7 = "Perl";
* const string L8 = "TypeScript";
* const string L9 = "Ruby";
* const string L10 = "Javascript";
* const string L11 = "SmallTalk";
* const string L12 = "Pascal";

**Cantidad de lenguajes por empleado:**

* E1: {L1; L2; L5; L6}
* E2: {L5; L6; L7; L8}
* E3: {L2; L3; L4}
* E4: {L9; L10; L11}
* E5: {L3; L4; L7; L8; L11; L12}
* E6: {L6; L7; L8; L10; L11; L12}
* E7: {L5; L9}
* E8: {L4; L8; L12}

Con estos valores de entrada el algoritmo aproximado devuelve una cantidad mínima de 3 empleados que cubren todos los lenguajes. Respondiendo con el resultado ln|8| +1.

Con los mismos valores de entrada el algoritmo backtracking devuelve una cantidad mínima de 3 empleados que cubren todos los lenguajes. A diferencia del algoritmo aproximado la eficiencia de este es de (#Hijos^#Niveles\*Costo) = (8^12 \* 8)