

## Inductive Analysis

La idea es sobreaproximar las variables inductivas haciendo un dependence analysis, usando como base las variables del header de los loops.

Va a ser un Backward Flow Analysis, porque....tengo que ir para atrás: una variable  $a$  depende de  $b$  en cierto program point si  $b$  fue usado *antes* para modificar el valor de  $a$ .

Hay que definir los siguientes objetos:

- genSet: se define para cada nodo (statement).
- killSet: ídem killSet
- in
- out
- Función de transferencia:  $\text{gen}(n) \cup (X \setminus \text{kill}(n))$
- Operación suma del reticulado

y probar que todo anda bien.

- genSet: el genSet de cada nodo (cada program point) va a ser vacío salvo que sea el header de un loop (o la creación de un array, que tratamos como el header de un loop)
- killSet: por alguna razón Diego lo define como vacío. Yo habría pensado que debería tener algo, pero quizás lo pienso porque me confundo con el Live Variable Analysis.
- La operación suma va a ser la unión, me parece, porque queremos quedarnos con todas las posibles variables inductivas.