Extensions to the Code Generator for Alpha

Inderaj S Bains
IRISA, Rennes

IRISA

Overview

- Introduction
  - What CodeGen does
  - Where does CodeGen fit in
- Previous Work
- Extensions Proposed

IRISA

Where Does Code Generator fit in :

What Code Generator does :

IRISA

Where Does Code Generator fit in :

What Code Generator does :

- Why is it needed :
  - ★ Checking correctness of program

IRISA

Previous Work

IRISA

Previous Work

- writeC( ) in C by Doran Wilde
  - ★ Code ⇒ Demand Drivens

IRISA

Previous Work

- writeC( ) in C by Doran Wilde
  - ★ Code ⇒ Demand Drivens
- CodeGen( ) in Mathematica by Fabien Quillere
  - ★ Code ⇒ Respecting Schedule

IRISA

Previous Work

- writeC( ) in C by Doran Wilde
  - ★ Code ⇒ Demand Drivens
- CodeGen( ) in Mathematica by Fabien Quillere
  - ★ Code ⇒ Respecting Schedule
- Areas untouched by CodeGen( )
  - ★ writeC( ) functionality
  - ★ Reductions
  - ★ Subsystems

IRISA

Functionality of writeC in CodeGen

IRISA

Functionality of writeC in CodeGen

- Motivations
  - ★ People used to
  - ★ Single Package
  - ★ Easier Maintainence and Extension

IRISA

Functionality of writeC in CodeGen

- Motivations
  - ★ People used to
  - ★ Single Package
  - ★ Easier Maintainence and Extension
- Handling
  - ★ CodeGen called Without Schedule or option noSched set
  - ★ different function called internally in same file

IRISA

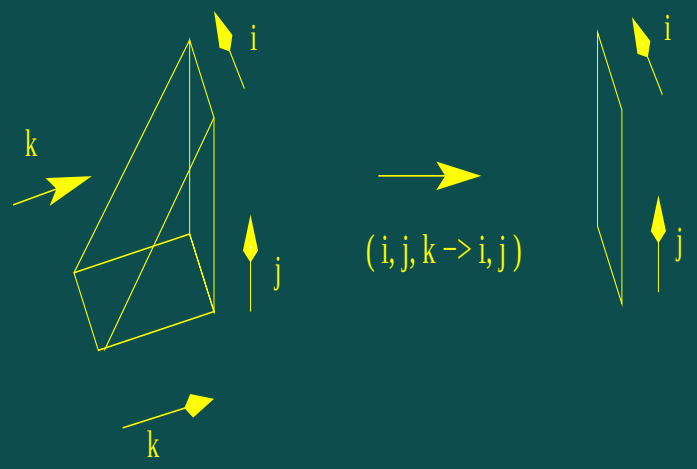
Handling Reductions

- What is a reduction :
  - reduce(op , affine ,expr)
  - B[i, j] = A[i, j-1] - reduce(+, (i, j,k $\longrightarrow$ i, j), Y)

IRISA

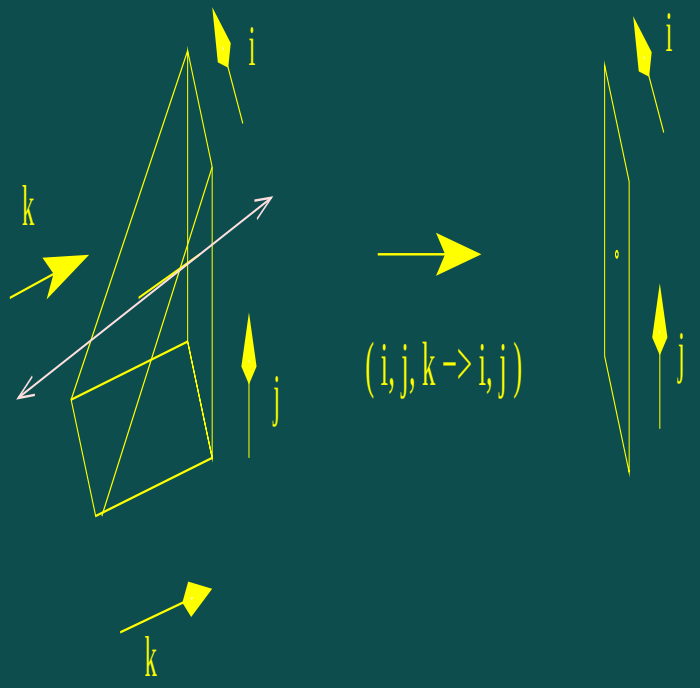
Handling Reductions

- What is a reduction :
  - reduce(op , affine ,expr)
  - B[i, j] = A[i, j-1] - reduce(+, (i, j,k ⟶ i, j), Y)
- physically. . .



IRISA

- How will it be handled:
  - We take preimage of point on expr domain by the affine function, intersect with the domain of expr and generate a loop for the resultant polyhedra



IRISA

Handling Subsystems

IRISA

Handling Subsystems

- What are subsystems :
  - ★ $\{$domain$\}$ use sub-name[ ] (inputs) returns (outputs);
  - ★ $\{i , j \mid 1 \leq i \leq j \leq N\}$ use Lxb[ ] (L , B) returns(X);

IRISA

Handling Subsystems

- What are subsystems :
  - ★ $\{$domain$\}$ use sub-name[ ] (inputs) returns (outputs);
  - ★ $\{i , j \mid 1 \leq i \leq j \leq N\}$ use Lxb[ ] (L , B) returns(X);
- How can they be handled :
  - ★ To be handled as function calls

IRISA

Handling Subsystems

- What are subsystems :
  - ★ $\{$domain$\}$ use sub-name[ ] (inputs) returns (outputs);
  - ★ $\{i , j \mid 1 \leq i \leq j \leq N\}$ use Lxb[ ] (L , B) returns(X);
- How can they be handled :
  - ★ To be handled as function calls
- For testing subsystems I have:
  - ★ Equivalent blocked and un-bloocked programs
  - ★ C programs which generate input,feed in different scanning orders and compare outputs

IRISA

Handling Subsystems

- What are subsystems :
  - ★ $\{$domain$\}$ use sub-name[ ] (inputs) returns (outputs);
  - ★ $\{i , j \mid 1 \leq i \leq j \leq N\}$ use Lxb[ ] (L , B) returns(X);
- How can they be handled :
  - ★ To be handled as function calls
- For testing subsystems I have:
  - ★ Equivalent blocked and un-bloocked programs
  - ★ C programs which generate input,feed in different scanning orders and compare outputs
- Usefulness to the Test Bench :

IRISA

Conclusion

- We Have seen how :
  - Single package will be made
  - writeC( ) functionality will be provided
  - Reductions and Subsystems will be handled