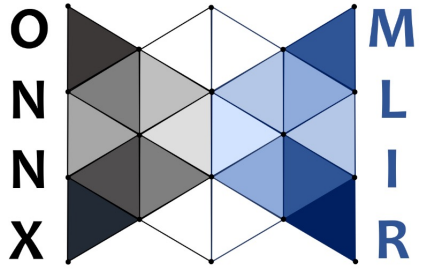
Dynamic Dimension Analysis in onnx-mlir Compiler

Tung D. Le (Speaker), Alexandre E Eichenberger, and Tong Chen

IBM Research



https://github.com/onnx/onnx-mlir

ONNX Community Meetup Day 2023

Dynamic Dimensions in Shape Inference

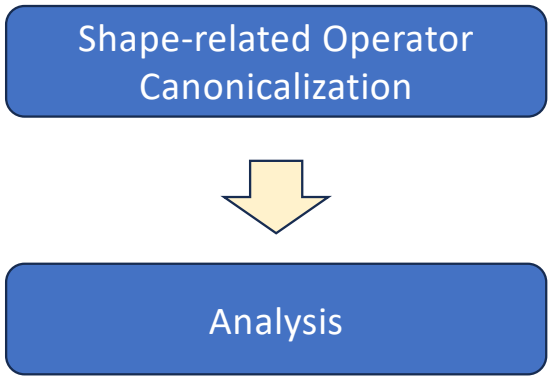
Dynamic dimension is an unknown value at compile time

```
"onnx.Add"(%A, %B) : (tensor<3x?x?xf32>, tensor<3x?x?xf32>) -> tensor<3x?x?xf32>
```

- Information about shapes is key to high performance
  - Ruling out broadcasting,
  - Enabling Fusion / optimized SIMD code patterns
  - Determining if an operation is suitable for hardware accelerator,…
- ONNX-provided shape inference is not enough
  - Compiler has shape inference anyway to generate code that computes dynamic shapes
  - Compiler changes ONNX operation patterns, has multiple dialects in addition to ONNX

Dynamic Dimension Analysis in onnx-mlir

- Purpose: to explore relations among dynamic dimensions
  - Current focus on discovering dimension equivalences
  - For shapes with arbitrary mixtures of static and dynamic dimensions
- Two phases

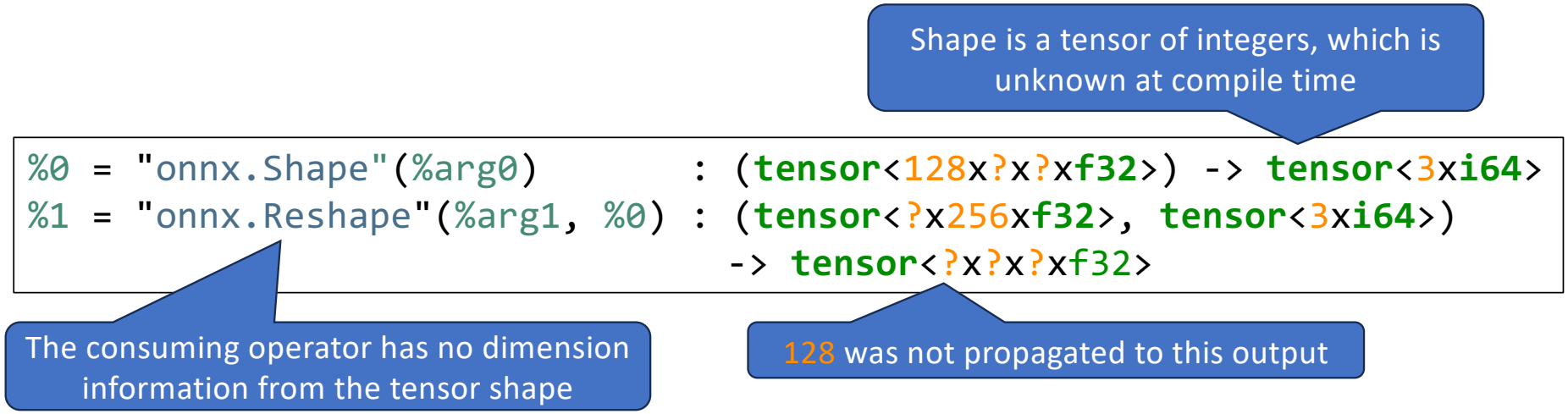


- Represent a shape by scalar constants and dimensions.
- Propagate scalars through shape-manipulating operators: Shape, Slice, Squeeze, etc.

- Explore dynamic dimension relationship using the existing shape inference infrastructure in onnx-mlir
- APIs to query information: sameDim(), sameShape()

Shape-related Operator Canonicalization (1/3)

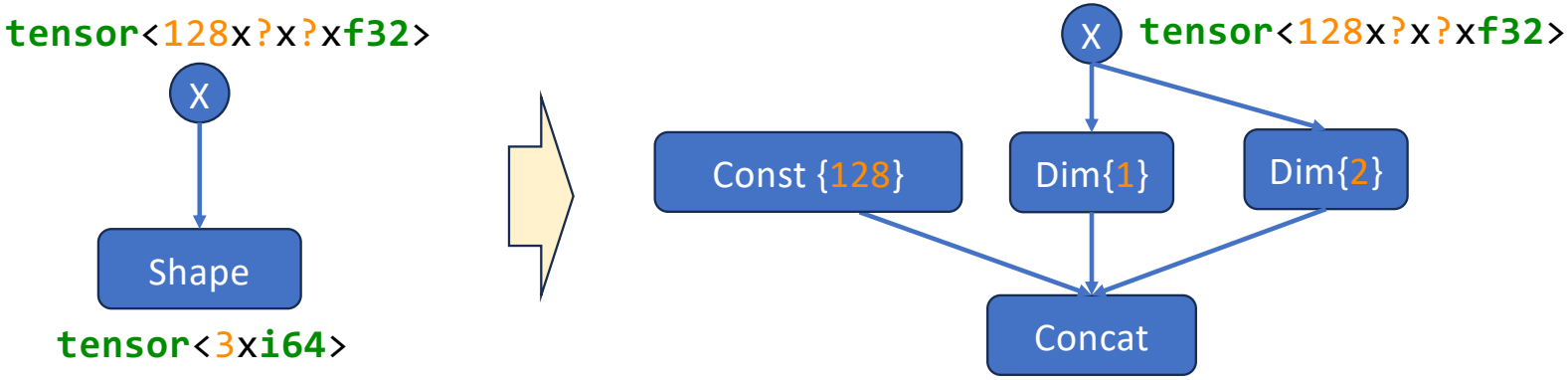
- onnx.Shape is often the starting point of shape calculation



- For complete shape inference, we must start keeping track of shape values that were saved in tensors.

Shape-related Operator Canonicalization (2/3)

- Expose individual dimensions to compiler

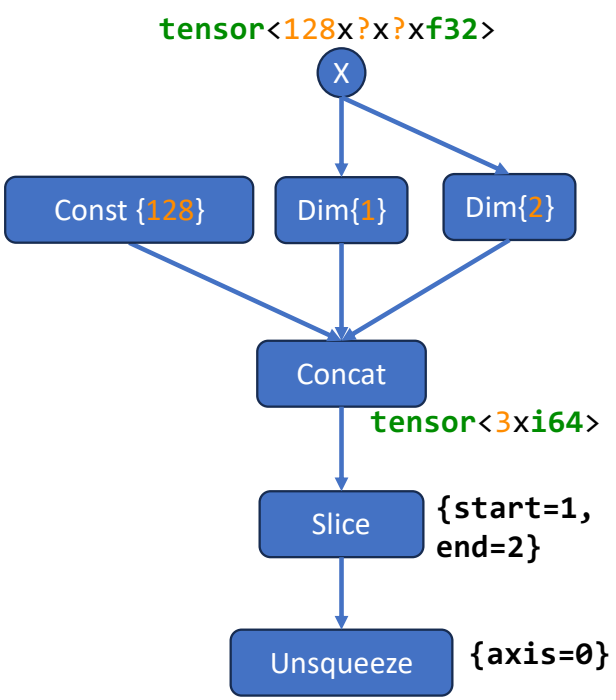


- A compile time shape value is represented by an onnx.Const
- A runtime shape value is represented by an onnx.Dim *

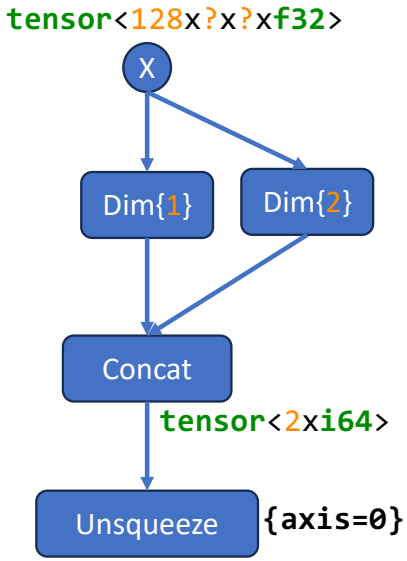
(*) onnx.Dim is an additional ONNX operator in onnx-mlir

Shape-related Operator Canonicalization (3/3)

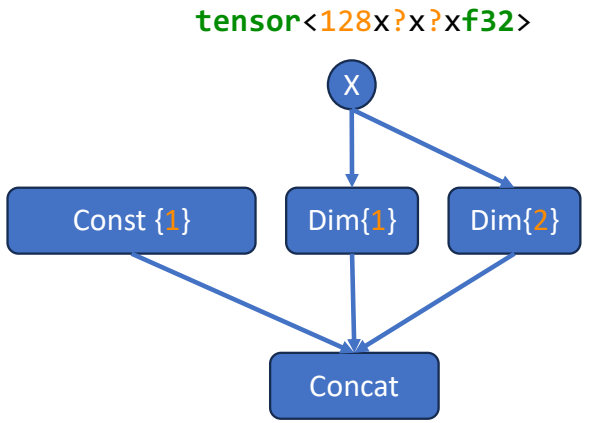
- Propagate scalar dimensions through shape-manupulating operators



Propagate through SliceOp

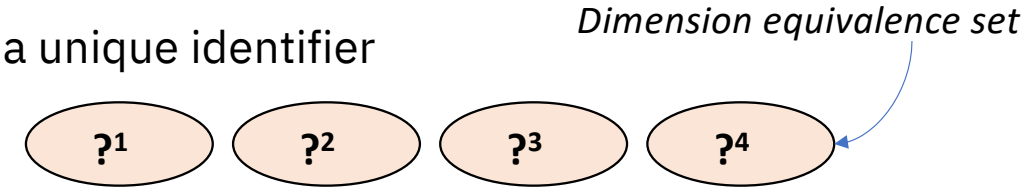


Propagate through UnsqueezeOp

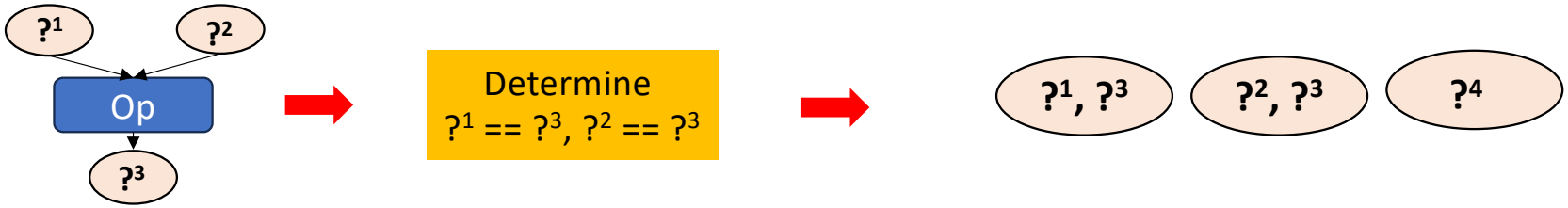


Analysis: Value Numbering for Shapes

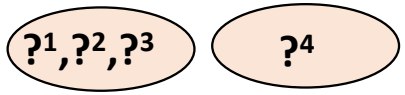
- Analysis algorithm:
  1. Associate each dynamic dimension `?` with a unique identifier



  2. Analyze an operation for dimension equivalence



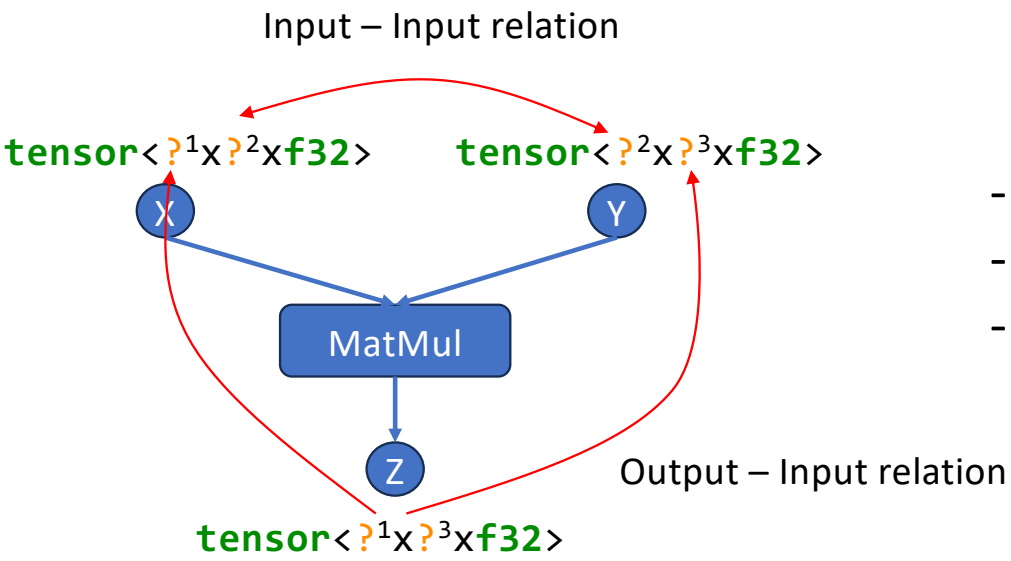
  3. Apply transitive closure of dimension equivalence



  4. Repeat Steps 2. & 3. for operations until steady state is reached

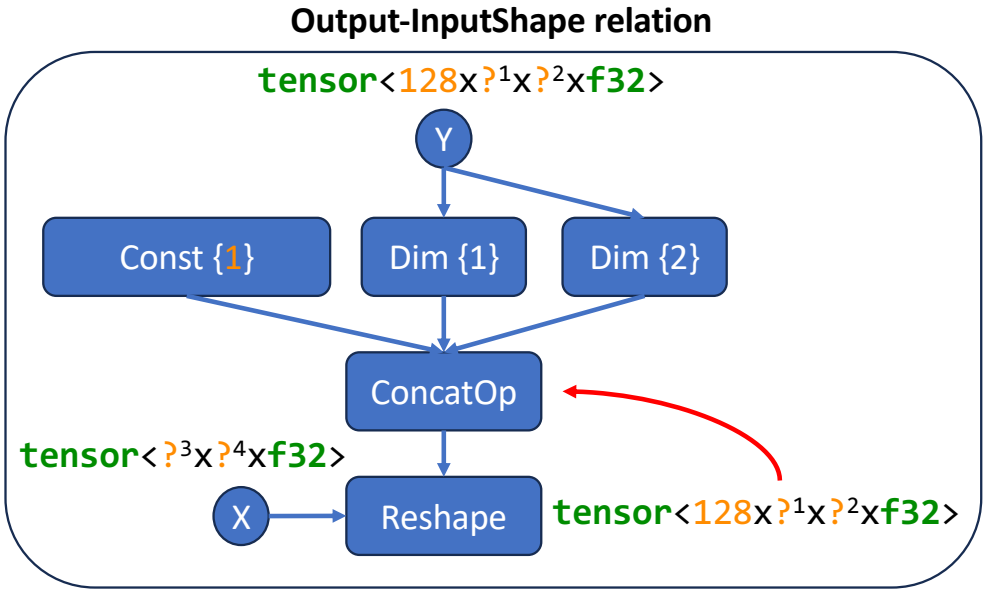
Expand a dimension set (1/2)

- Given an operator and a dynamic dimension in the output tensor
- Find ALL equivelant dynamic dimensions in the input tensors.



- Dim(Z, 0) is equivalent to Dim(X, 0)
- Dim(Z, 1) is equivalent to Dim(Y, 1)
- Dim(X, 1) is equivalent to Dim(Y, 0)

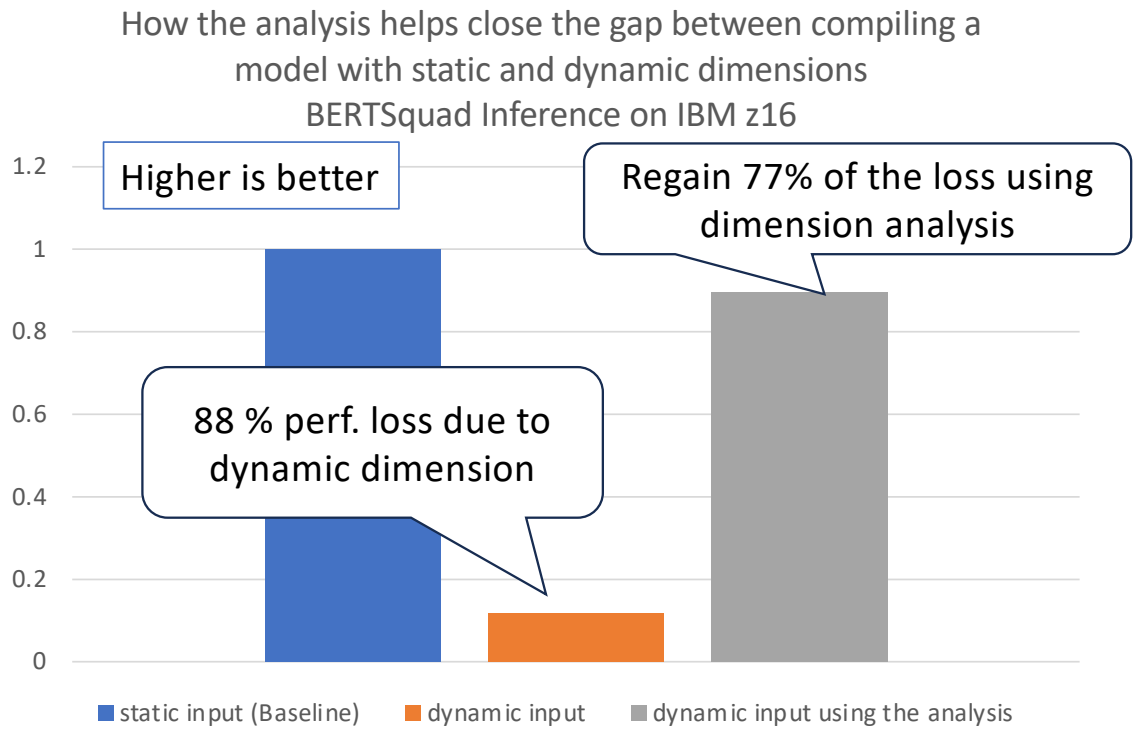
Expand a dimension set (1/2)



- We have implemented this rule for onnx Reshape, ConstantOfShape, Expand, MaxUnpool, CenterCropPad, Tile.

Apply to the BERTSquad model

- 854 dynamic dimensions are classified into 26 sets.
- Sizes of the 26 sets
  - 1 set of 817 dimensions
  - 12 sets with 2 dimensions
  - 13 sets with 1 dimension



- Future work:
  - Analysis with the presence of +, -, *, /

Thank you for your listening!