

TableGen BOF

Mihail Popa

The BoF was attended by quite a lot of people and there was general agreement that tablegen needs improvement in some shape or form. However there are many divergent ideas as to how to go about this improvement. Of course this is completely natural, tablegen being a versatile tool used by many different people for many different purposes.

Part of the attendees focused on missing features. Feature requests include:

1. support for vector literals
2. support for instruction/patterns with multiple outputs
3. support complex transforms where types of input and output differ
4. remove need of magic vars in complex patterns
5. improve support for permutations of operands
6. ability to implement DAG combines in tablegen
7. auto detect/infer constraints
8. add support or missing patterns and/or selectable insts
9. extend tablegen unit tests to go beyond simple syntactic constructs
10. automatically add MCDesc fields
11. improve debug-ability and error reporting
12. auto-inversion for predicates

It was commented that the current state of tablegen is due to its quite rapid growth through the past years. Then point was raised that, probably, we will not be fixing anything by simply bolting on top further functionality.

My desire was to steer the discussion towards how to design tablegen as a proper domain specific language targeted at compiler construction. I think it was generally agreed that this is a distant goal worth pursuing and that the first steps is to first document what tablegen currently has.

The root cause for lack of documentation was identified as:

"People don't feel authoritative enough to go ahead and write docs. Many community members are knowledgeable in small pieces, but don't feel are qualified to author documentation".

It was generally agreed that a way to solve this is to create a shared repository where everyone can commit any pieces of information they might have. We hope it will grow to a proper documentation for tablegen's features. Once gotten there we would be in a much better position to decide how to further develop this tool.