

git_comments:

1. ! * \brief Check whether the two expressions are equal or not, if not simplify the expressions and check again * \note This is stronger equality check than `tvm::ir::Equal` * * \param lhs First expression * \param rhs Second expression * * \return result True if both expressions are equal, else false

git_commits:

1. **summary:** Added equality check and upgraded concatenate op (#1172)
message: Added equality check and upgraded concatenate op (#1172)

github_issues:

1. **title:** [TOPI] Better Equality Checking
body: Currently TOPI relies on several functions to check equality constraint that are not in particularly general and produces problems in cases like concatenate #1078 #1024 We need to solve this issue, there are two orthogonal ways - [] Upgrade concatenate to simplify its output expression, so that it won't have composed sum - [] Define a stronger version of `topi::EqualCheck` that first calls `ir::Equal`, if failed, do `CanonicalSimplify(x - y)` and check if result equals 0 - [] Add regression test cases to prevent that from happening
2. **title:** [TOPI] Better Equality Checking
body: Currently TOPI relies on several functions to check equality constraint that are not in particularly general and produces problems in cases like concatenate #1078 #1024 We need to solve this issue, there are two orthogonal ways - [] Upgrade concatenate to simplify its output expression, so that it won't have composed sum - [] Define a stronger version of `topi::EqualCheck` that first calls `ir::Equal`, if failed, do `CanonicalSimplify(x - y)` and check if result equals 0 - [] Add regression test cases to prevent that from happening
3. **title:** [TOPI] Better Equality Checking
body: Currently TOPI relies on several functions to check equality constraint that are not in particularly general and produces problems in cases like concatenate #1078 #1024 We need to solve this issue, there are two orthogonal ways - [] Upgrade concatenate to simplify its output expression, so that it won't have composed sum - [] Define a stronger version of `topi::EqualCheck` that first calls `ir::Equal`, if failed, do `CanonicalSimplify(x - y)` and check if result equals 0 - [] Add regression test cases to prevent that from happening
4. **title:** [TOPI] Better Equality Checking
body: Currently TOPI relies on several functions to check equality constraint that are not in particularly general and produces problems in cases like concatenate #1078 #1024 We need to solve this issue, there are two orthogonal ways - [] Upgrade concatenate to simplify its output expression, so that it won't have composed sum - [] Define a stronger version of `topi::EqualCheck` that first calls `ir::Equal`, if failed, do `CanonicalSimplify(x - y)` and check if result equals 0 - [] Add regression test cases to prevent that from happening
label: test
5. **title:** [TOPI] Better Equality Checking
body: Currently TOPI relies on several functions to check equality constraint that are not in particularly general and produces problems in cases like concatenate #1078 #1024 We need to solve this issue, there are two orthogonal ways - [] Upgrade concatenate to simplify its output expression, so that it won't have composed sum - [] Define a stronger version of `topi::EqualCheck` that first calls `ir::Equal`, if failed, do `CanonicalSimplify(x - y)` and check if result equals 0 - [] Add regression test cases to prevent that from happening
label: code-design
6. **title:** [TOPI] Better Equality Checking
body: Currently TOPI relies on several functions to check equality constraint that are not in particularly general and produces problems in cases like concatenate #1078 #1024 We need to solve this issue, there are two orthogonal ways - [] Upgrade concatenate to simplify its output expression, so that it won't have composed sum - [] Define a stronger version of `topi::EqualCheck` that first calls `ir::Equal`, if failed, do `CanonicalSimplify(x - y)` and check if result equals 0 - [] Add regression test cases to prevent that from happening
7. **title:** [TOPI] Better Equality Checking

body: Currently TOPI relies on several functions to check equality constraint that are not in particularly general and produces problems in cases like concatenate #1078 #1024 We need to solve this issue, there are two orthogonal ways - [] Upgrade concatenate to simplify its output expression, so that it won't have composed sum - [] Define a stronger version of `topi::EqualCheck` that first calls `ir::Equal`, if failed, do `CanonicalSimplify(x - y)` and check if result equals 0 - [] Add regression test cases to prevent that from happening

label: code-design

8. **title:** [TOPI] Better Equality Checking

body: Currently TOPI relies on several functions to check equality constraint that are not in particularly general and produces problems in cases like concatenate #1078 #1024 We need to solve this issue, there are two orthogonal ways - [] Upgrade concatenate to simplify its output expression, so that it won't have composed sum - [] Define a stronger version of `topi::EqualCheck` that first calls `ir::Equal`, if failed, do `CanonicalSimplify(x - y)` and check if result equals 0 - [] Add regression test cases to prevent that from happening

label: code-design

github_issues_comments:

1. @PariksheetPinjari909 Are you interested in help on this?
2. @PariksheetPinjari909 can you verify if the case still exists, if so apply the listed fixes?
3. Sorry, i missed your comment. Of course i will be interested in working on this. Yes the issue still exist and we need one of the solution provided by you above. What i feel is we can simplify the output expression there itself in concatenate operation. Could you throw more light on 3rd point of regression test. In the mean time i will prepare the PR for concatenate op upgrade.
4. **body:** by regression test I mean build a testcase that fails in current master but can be fixed in the patch.
label: test
5. **body:** Both the issues #1078 #1024 are caused in CPP code. When i tested it on python equivalent, it passed fine. But keeping consistency in mind, i suggest we should make changes in both cpp and python concatenate function to simplify expression. What are your views on this @tqchen ?
label: code-design
6. sound sgood, we should also add a stronger version of `EqualCheck` in C++

github_pulls:

1. **title:** Added equality check and upgraded concatenate op
body: This PR solves the issues referred in #1106.

github_pulls_comments:

github_pulls_reviews:

1. add \note This is a stronger check than `tvm::ir::Equal`

jira_issues:

jira_issues_comments: