Enable TCO by use of special form

06/10/2020 09:38 AM - x3qt (Yury Polyakov)

I would like to propose to create some way to enable TCO in ruby by use of special form, like (recur)\[https://clojure.org/reference/special_forms#recur\].

AFAIK TCO is not enabled by default because it messes with stack traces, so by implementing some special form we will avoid changing default behavior and provide opportunity to use it at the same time.

Example:

def fact(n, acc = 1)
  return acc if n <= 1
  recur(n - 1, n * acc) # recursive call with TCO enabled
end

Related issues:
Is duplicate of Ruby master - Feature #12543: explicit tail call syntax: foo(...
foo() then return

Syntax proposed in #12543 does not look good to me, specifically because it is not a single word and using already existing constructions, so can be confusing because it is a special meaning.

Adding a new reserved word is extremely expensive in the sense that it massively breaks backwards-compatibility. Every single piece of code everywhere in the world that every programmer has ever written that uses the identifier recur will be potentially broken by this change.

The code you proposed is perfectly legal today, which means that introducing your proposed change will change the meaning of existing code.

If you want to introduce a feature for adding new return semantics to methods you have to do it in a way which

- does not introduce a new keyword or reserved word
- does not change the meaning of existing code
- ideally is currently not valid Ruby code

foo() then return

has the advantage that it only uses already existing reserved words and keywords and is currently illegal, so there cannot possibly be any existing code that gets broken.

#8 - 04/19/2021 12:21 AM - nobu (Nobuyoshi Nakada)
- Status changed from Open to Closed

Close due to duplication.