CIT plugin and Avocado framework installation guide

The CIT plugin is a part of Avocado framework.

First make sure you have a basic set of packages installed. The following applies to Fedora based distributions, please adapt to your platform:

```
sudo dnf install -y python3 git gcc python3-devel python3-pip libvirt-devel libffi-devel openssl-devel libyaml-devel redhat-rpm-config xz-devel
```

Then for installation run:

```
$ cd avocado
$ sudo make requirements
$ sudo python3 setup.py install
```

Now you should have installed Avocado framework and CIT plugin.

For better understanding of Avocado framework you can read the documentation:

https://avocado-framework.readthedocs.io/en/latest/

The code of CIT plugin is in avocado/optional_plugins/varianter_cit and you can run and example with

```
avocado variants --cit-parameter-file avocado/examples/varianter_cit/params.cit
```

CIT plugin guide:

Input file format

The following is the general structure of a input file:

```
PARAMETERS
Parameter_1 [Value_1,  Value_2, Value_3, Value_4]
Parameter_2 [Value_1,  Value_2, Value_3, Value_4]
Parameter_3 [Value_1,  Value_2, Value_3, Value_4]

CONSTRAINTS
Parameter_1 != Value_1 || Parameter_2 != Value_3
Parameter_3 != Value_2 || Parameter_2 != Value_4 || Parameter_1 != Value_4
```

The input file has two parts, parameters and constraints.

Parameters:

- Each line represent one parameter.
- Each parameter has a name, and a list of values inside brackets.

Constraints:

- Constraints have to be in Conjunctive normal form.
- Constraints use these tree operands: `!=, OR, AND`
- `||` represents operand `OR` and new line represents operand `AND`.
- **In the example is this logic formula::**

  ((P_1 != V1 OR P_2 != V_3) AND (P_3 != V_2 OR P_2 != V_4 OR P_1 != Value_4))

Usage

the algorithm employed here can be CPU intensive. If you want more information on the progress of the combinatorial calculation, add `--debug` to a command line, such as `avocado variants –debug --cit-parameter-file $PATH`

Cit varianter plugin runs with three parameters:

- –cit-parameter-file with path to the input file
- –cit-order-of-combinations with strength of combination (default is 2)
- --cit-cycle-number with number of CIT iterations (default is 600)

You can enable more verbosity, making each variant to show its content:

```
avocado variants --cit-parameter-file avocado/examples/varianter_cit/params.cit -c
```

To execute tests with CIT combinations use:

```
avocado run passtest.py --cit-parameter-file avocado/examples/varianter_cit/params.cit
```