

Достаточно выяснить, не вызывается данная ячейка в формуле в ней самой. Остальные ячейки будут проверяться при их Set(...)

Cell::HasCyclic(const Cell* target, const Cell* current, std::unordered_set<const Cell*>& visited)
target - это всегда this, из которого первоначально вызывается метод. В аргументах нужен для передачи по цепочке рекурсивных вызовов
В current будет текущая ячейка для сравнения с target. Изначально в неё подадим временную ячейку, создаваемую при вызове Set(...) и существующую только для этой проверки и передачи данных после либо выкидывания исключения и сохранения изначальных данных

Проверяем, формула ли в ячейке через хелпер

Если Формула

Нет

return false
Нет формулы - нет циклических зависимостей

Да

for(const Position& position : formula->GetReferences())
Cell* temp = sheet_.GetCell(position);

if(temp == target)
Если текущая ячейка совпадает с начальной

Да

return true
Возвращаем true и кидаем исключение в Set()

Опытным путём выяснил что сначала нужно проверить на равенство, а потом - смотреть в visited так как мы добавляем наш target в visited в первой итерации

if(visited.count(temp) == 0)
Проверка на то, проверяли ли мы уже эту ячейку

Нет

continue

visited.insert(temp);
Рекурсивно перебираем все ячейки, от которых зависим
if(temp->HasCyclic(target, temp, visited))

Если хоть в одной да

return true
Возвращаем true и кидаем исключение в Set()

Во всех нет

return false;
Можно подставлять новую формулу в ячейку