```
from rdflib import Graph, Namespace, RDF, RDFS, OWL, XSD, Literal


# Define paths and file names
BASE = 'http://ex.org#'
ONT_FILE = 'ontology.ttl'
PAPER_FILE = 'papers.txt'
AUTHOR_FILE = 'authors.txt'
KG_FILE = 'created_kg.ttl'


# Create rdflib graph
Graph()

# Read ontology from file
g.parse(ONT_FILE, format='owl')


def str_to_id(str):
    """Make sure that the string is a legal part of a URL."""
    return str.replace(' ', '_')


def add_paper(g, line):
    """Add information about a paper to the graph."""
    if line.strip()=='0':
        return
    title, author_names, source, pub_name, year = line.split(';')

    paper = BASE['publ_'+str_to_id(title.strip())]
    g.add(paper, RDF.type, BASE.Paper)
    g.add(paper, BASE.title, title.strip())
    g.add(paper, BASE.year, int(year))

    publication = BASE['publ_'+str_to_id(source.strip())]
    g.add(publication, BASE.title, source.strip())
    g.add(paper, BASE.publication, publication)

    publisher = BASE['org_'+str_to_id(pub_name.strip())]
    g.add(publisher, BASE.name, pub_name.strip())
    g.add(paper, BASE.publisher, publisher)

    for auth_name in author_names.split(','):
        author = BASE['author_'+str_to_id(auth_name.strip())]
        g.add(author, BASE.name, auth_name.strip())
        g.add(paper, BASE.author, author)


# Read paper information from file and add to graph
with open(PAPER_FILE) as file:
    for line in file:
        add_paper(g, line)


def add_author(g, line):
    """Add information about an author to the graph."""
    if line.strip()=='0':
        return
    name, aff_name, country_name = line.split(';')

    author = BASE['author_'+str_to_id(name.strip())]
    g.add(author, RDF.type, BASE.Author)
    g.add(author, BASE.name, name.strip())

    affiliation = BASE['org_'+str_to_id(aff_name.strip())]
    g.add(affiliation, BASE.name, aff_name.strip())
    g.add(author, BASE.affiliation, affiliation)

    country = BASE['country_'+str_to_id(country_name.strip())]
    g.add(country, BASE.name, country_name.strip())
    g.add(author, BASE.country, country)


# Read author information from file and add to graph
with open(AUTHOR_FILE) as file:
    for line in file:
        add_author(g, line)

# Execute RDFS entailments
rdfs = RDFS_OWLRL_Semantics(g, False, False, False)

# Remove unnecessary triples
g.update("""
DELETE {
    ?s ?p rdfs:Resource .
}
""")

# Save to file
g.serialize(KG_FILE, format='ttl')
```