Inspect your runnables

Once you create a runnable with LCEL, you may often want to inspect it to get a better sense for what is going on. This notebook covers some methods for doing so.

First, let's create an example LCEL. We will create one that does retrieval

```
%pip install --upgrade --quiet  langchain
langchain-openai faiss-cpu tiktoken
```

```
from langchain.prompts import
ChatPromptTemplate
from langchain.vectorstores import FAISS
from langchain_core.output_parsers import
StrOutputParser
from langchain_core.runnables import
RunnableLambda, RunnablePassthrough
from langchain_openai import ChatOpenAI,
OpenAIEmbeddings
```

```
vectorstore = FAISS.from_texts(
    ["harrison worked at kensho"],
embedding=OpenAIEmbeddings()
)
retriever = vectorstore.as_retriever()

template = """Answer the question based only
on the following context:
{context}

Question: {question}
"""
prompt =
ChatPromptTemplate.from_template(template)

model = ChatOpenAI()
```

```
chain = (
    {"context": retriever, "question":
RunnablePassthrough()}
    | prompt
    | model
    | StrOutputParser()
)
```

Get a graph

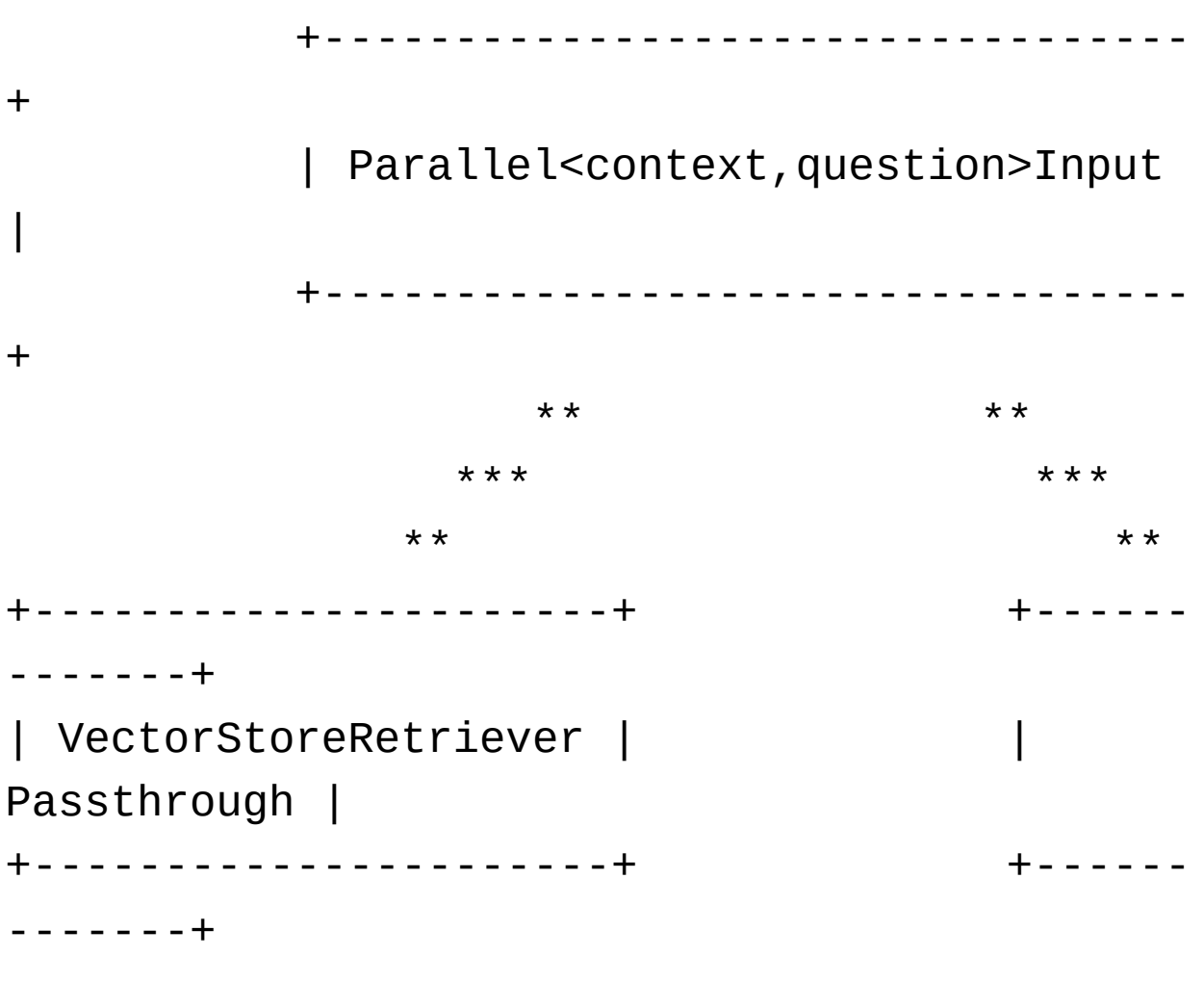
You can get a graph of the runnable

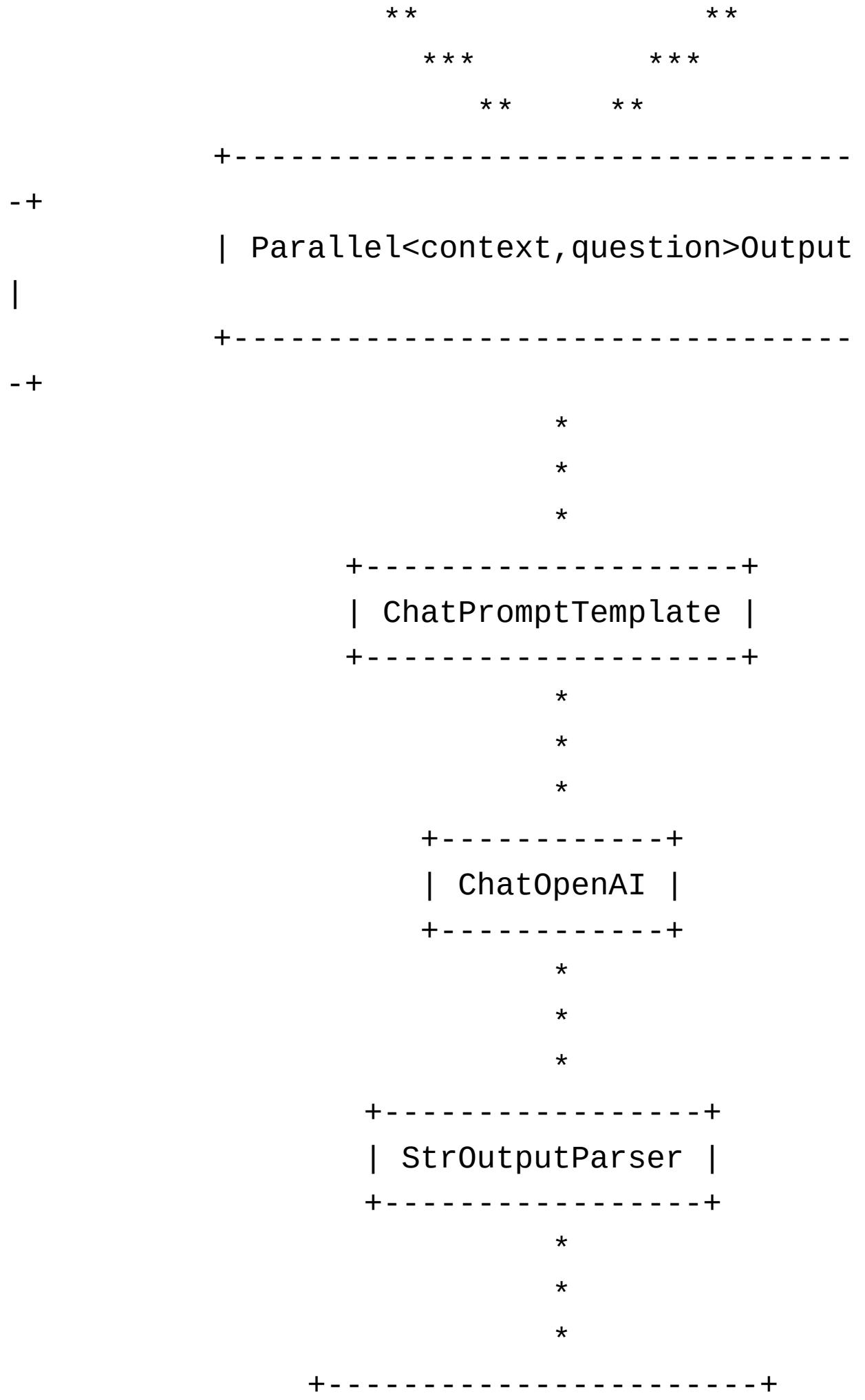
```
chain.get_graph()
```

Print a graph

While that is not super legible, you can print it to get a display that's easier to understand

```
chain.get_graph().print_ascii()
```




Parallel<context,question>Output
ChatPromptTemplate
ChatOpenAI
StrOutputParser

```
                    | StrOutputParserOutput |
                    +-----------------------+
```

Get the prompts

An important part of every chain is the prompts that are used.

You can get the prompts present in the chain:

```
chain.get_prompts()
```

```
[ChatPromptTemplate(input_variables=['context',
messages=
[HumanMessagePromptTemplate(prompt=PromptTempla
['context', 'question'], template='Answer the q
on the following context:\n{context}\n\nQuestio
{question}\n'))])]
```