**سوال تئوری:**

روش اول model checking :

در این روش برای درستی یک گزاره با توجه به KB های ما، تمام حالت های ممکن را بررسی میکنیم. برای مثال در بازی Wumpus world اگر بخواهیم بدانیم در یک خانه ی خاص آیا Wumpus وجود دارد یا نه. برای این کار باید تمام حالات را بررسی کرد و درصورت وجود conflict این جمله غلط و در غیر این صورت گزاره ما درست است.

روش دوم theorem-proving:

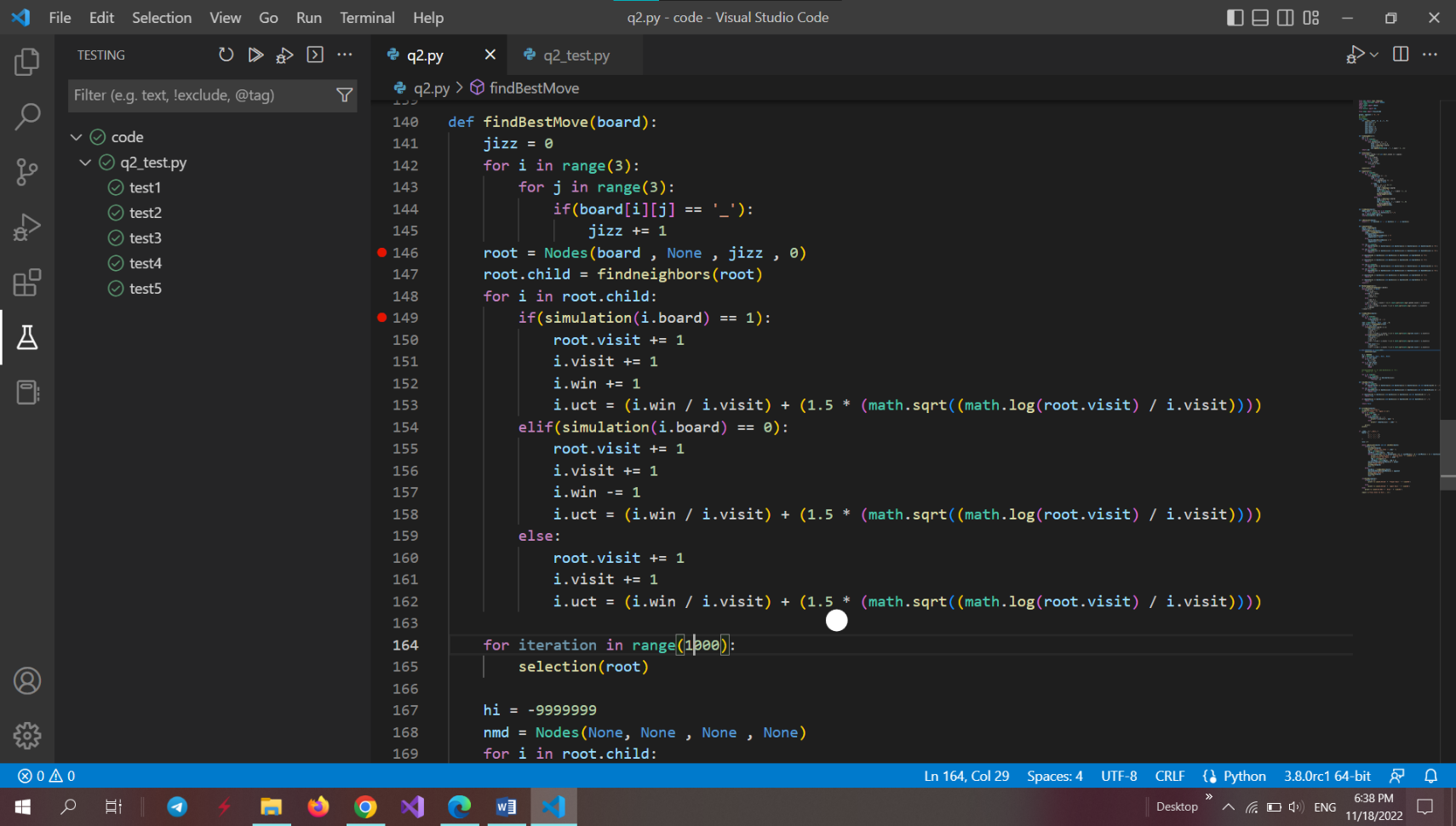
در این روش بجای بررسی تمام حالات برای درستی یک گزاره، با استفاده ازKB ها و قوانین استنباط، نتیجه گیری های جدید انجام میدهیم.مثلا با استفاده ازmodus ponens میدانیم اگر دریک خانه نسیم بیاید آنگاه در خانه های همسایه آن چاه داریم. حالا اگر در خانه ای که هستیم نسیم نیاید پس در خانه های همسایه آن چاه وجود ندارد.

**سوال عملی:**

برای بازی در ایکس او پس از انتخاب حرکت توسط player حرکت بعدی باید توسط تابع findBestMove که مشخصات خانه ای که O در آن قرار میگیرد به عنوان خروجی داده شود.

برای این کار باید ابتدا یک ریشه که وضعیت فعلی جدول است را بسازیم. سپس هربار 4 مرحله را انجام دهیم :

ابتدا با استفاده از selection نودی که بیشترین ucb را دارد و همچنین تمام نود های چالد آن ساخته شده و leaf نیست، انتخاب میکنیم. در مرحله بعد نود انتخاب شده را باید expand کنیم. به این شکل که یکی از نود های چایلد آن را بسازیم. حال باید با استفاده از simulation یک بار بازی را کامل انجام دهیم به شکلی که حالت اولیه ی جدول حالتی که این نود است، باشد.سپس با استفاده از Backpropagation مقدار win , visit , ucb تمام نود ها را تغییر بدهیم به این شکل که اگر نوبت player بود، و ما در simulation امان opponent برد، از win در این نود 1 کم شده اگر نوبت opponent بود، به مقدار winنود 1 اضافه میشود و در صورتی مساوی شدن مقدار win تغییر نکرده و visit یکی اضافه میشود. این مراحل 1000 بار انجام داده میشود.

درآخر تمام چالد های ریشه را چک کرده و نودی که بیشترین ucb دارد را به عنوان حرکت بعدی انتخاب میکنیم.