پروژه درس ساختمان داده

تعریف مسئله:

همانطور که از درس ساختمان گسسته به یاد داریم، گزاره یک جمله خبری است (جمله‌ای که یک fact را ارائه می‌دهد) و یا درست است یا غلط ولی نه هر دو. گزاره‌هایی که بر حسب گزاره‌های کوچکتر ساخته نشده‌اند، گزاره‌های اتمیک نام دارند. گزاره‌هایی اتمیک را می‌توان با عملگرهای منطقی با هم ترکیب کرد و گزاره‌های مرکب را ساخت. یکی از روش‌های رایج بررسی ارزش گزاره‌ها جدول درستی است.

مثال ۱) گزاره $\neg(p \wedge q)$ از دو گزاره اتمیک p و q ساخته شده است. جدول درستی این گزاره به صورت شکل ۱ است:

p	q	$p \wedge q$	$\neg(p \wedge q)$
F	F	F	T
F	T	F	T
T	F	F	T
T	T	T	F

شکل ۱-جدول درستی گزاره $\neg(p \wedge q)$

یک گزاره S-type نامیده می‌شود اگر حداقل یک ترکیب از گزاره‌های اتمیک وجود داشته باشد که به ازای آن ارزش گزاره درست (True) باشد.

یک گزاره V-type نامیده می‌شود اگر همواره ارزش آن True باشد.

تشخیص S-type بودن گزاره‌ها نقش بسیار مهمی در reasoning (استدلال) دارد.

هدف اصلی این پروژه تشخیص گزاره‌های S-type و V-type است.

ورودی/خروجی مسئله:

ورودی/خروجی مسئله:

ابتدا گزاره‌های اتمیک از کاربر گرفته می‌شود. در مثال فوق، گزاره‌های اتمیک p و q به عنوان ورودی دریافت می‌شوند.

سپس گزاره اصلی از کاربر دریافت می‌شود. این گزاره متشکل از گزاره‌های اتمیک و پنج عملگر منطقی $\neg \leftrightarrow \rightarrow \vee \wedge$ است. در مثال فوق، $\neg(p \wedge q)$ توسط کاربر وارد می‌شود.

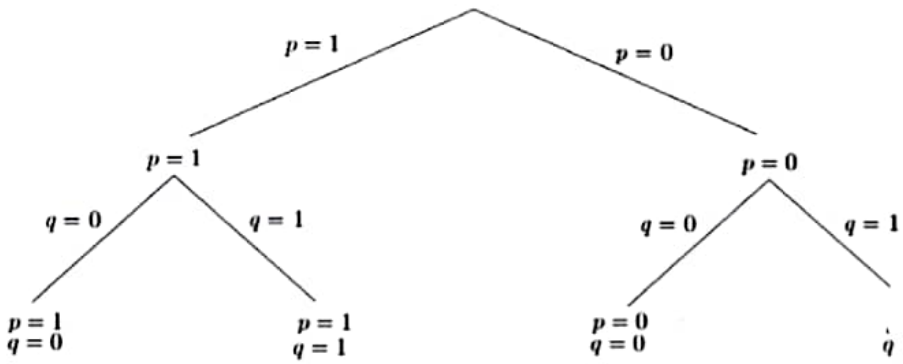
در نهایت کاربر نوع مورد نظر (S-type, V-type) را مشخص می‌کند.

برنامه مشخص می‌کند که آیا گزاره از نوع مشخص شده است یا خیر.

- در صورتی که هدف S-type باشد: اگر گزاره از این نوع بود، عبارت Yes یک ترکیب که باعث مقدار درستی True می‌شود، در خروجی چاپ می‌گردد. اگر گزاره از این نوع نبود، پیغام It is not a S-type proposition چاپ می‌شود.
- در صورتی که هدف V-type باشد: اگر گزاره از این نوع نبود، عبارت No و یک ترکیب که باعث مقدار False می‌شود، در خروجی چاپ می‌گردد. اگر گزاره از این نوع بود، پیغام It is a V-type proposition چاپ می‌شود.

حل مسئله:

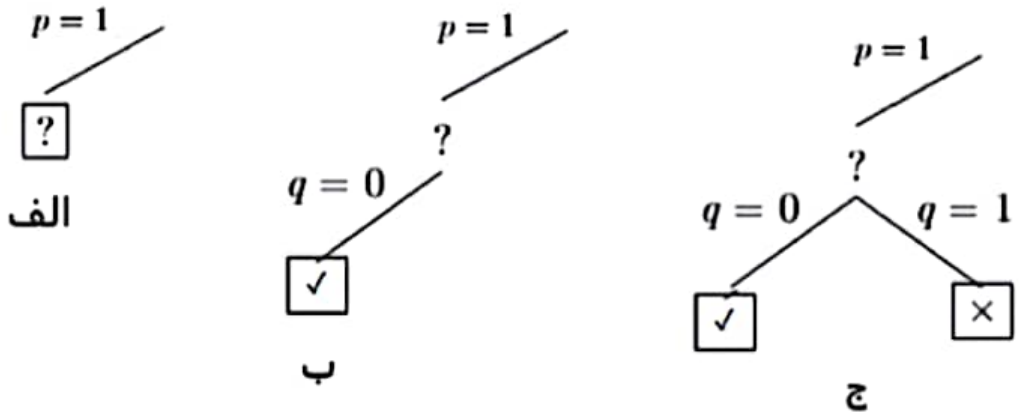
روش جدول درستی Brute-force است زیرا تمام حالات را بررسی می‌نماید. این روش تنها برای موارد بسیار کوچک عملی است. زیرا با افزایش تعداد گزاره‌ها، تعداد سطرهای جدول به صورت نمایی رشد می‌کنند. یک راه‌حل بهتر، مدل کردن فضای جستجو به صورت درخت و پیمایش آن به صورت عمقی است. به عنوان مثال فضای جستجو در مثال فوق به صورت شکل ۲ خواهد بود. توجه کنید همه حالات ممکن از گزاره‌های اتمیک در برگ‌ها قرار گرفته‌اند.



شکل ۲-فضای جستوی گزاره مثال ۱ به فرم درختی

در جستجوی اولیه به صورت backtracking، از ریشه درخت شروع می‌کنیم (تخصیص تهی که هیچ چیز مشخص نیست) و در راستای شاخه به سمت پایین حرکت می‌کنیم. در هر گره، بررسی می کنیم که آیا گزاره X درست است یا خیر. اگر X فعلا درست باشد، با انتخاب **یک مقدار برای گزاره‌ی اتمیکی که در حال حاضر اختصاص داده نشده است**، در شاخه فعلی به سمت پایین حرکت می کنیم. اگر در گره‌ای تشخیص دهیم مقدار X، بدون توجه به سایر گزاره‌های مقدار نگرفته در آن شاخه، نادرست/درست است، آن شاخه هرس شده و دیگر ادامه نمی‌دهیم. با توجه به نوع گزاره مورد بررسی، در صورت لزوم به گره والد برگشته و شاخه دیگر را بررسی می‌کنیم.

مثال ۱ را در نظر بگیرید. گزاره هدف $X = \neg(p \wedge q)$ است. در شکل ۳، در هر مرحله، بخشی از درخت را نشان می دهیم که تاکنون کاوش شده است. علامت × در یک گره نشان می دهد که گزاره X در این شاخه نادرست خواهد بود. مقدار ✓ نشان می‌دهد که گزاره X در این شاخه درست خواهد بود. مقدار ؟ نشان می دهد که در حال حاضر در آن گره مقدار X مشخص نیست. اگر هدف بررسی S-type بودن X باشد، در شکل ۳ قسمت ب به جواب رسیده، جستجو متوقف شده و خروجی $Yes: (p = 1, q = 0)$ چاپ می‌شود. اگر هدف بررسی V-type بودن گزاره باشد، در شکل ۳ قسمت ج به جواب رسیده، جستجو متوقف شده و خروجی $No: (p = 1, q = 1)$ چاپ می‌شود.



شکل ۳-پیمایش درخت برای گزاره مثال ۱

نکات: ساختمان داده درخت باید به صورت کامل پیاده‌سازی شود. در صورت استفاده از ساختمان داده پشته برای backtracking در درخت، این ساختمان داده نیز باید کامل پیاده سازی شود.