

## آزمون میانی درس طراحی کامپایلر

مجموع نمره‌ها: ۱۰۰

زمان آزمون: ۸۰ دقیقه

۱ (۱۰) در مرحله‌ی تحلیل واژه‌ای (Lexical Analysis) به کمک عبارات منظم، برخی از واژه‌ها با دو الگوی زبان منطبق می‌شوند (برای نمونه، کلمه‌ی رزرو شده‌ی «for» با الگوی نام یک متغیر نیز منطبق می‌شود). علاوه بر این مسئله، گاهی واژه‌ها و پیشوندهای (Prefix) آنها، با الگوهای متفاوتی منطبق می‌شوند (برای مثال پیشوند «for» از واژه‌ی «form» با کلمه‌ی رزرو شده‌ی «for» منطبق می‌شود). برای حل این دو مشکل، چه راهکاری معمولاً اتخاذ می‌شود؟

۲ (۲۰) در زبان C اعداد ثابت در برنامه چند شکل می‌توانند داشته باشند: در صورتی که عددی با حروف 0x شروع شود، در مبنای شانزده، در صورتی که با حرف 0 شروع شود، در مبنای هشت و در غیر این صورت، عدد در مبنای ده می‌باشد. عبارت منظمی برای آن ارائه دهید. عبارت منظمی که ارائه می‌دهید برای مثال باید با 023، 0x1fa و 129 منطبق شود ولی نباید با 019 و 1fb منطبق شود.

۳ (۲۰) برای گرامر زیر، شبه کد یک تجزیه‌گر «Recursive Descent» را ارائه دهید. فرض کنید تابع (`nexttok()`) واژه‌ی بعدی را بر می‌گرداند و تابع (`droptok()`) واژه‌ی بعدی را دور می‌اندازد. نمادهای پایانی (Terminal) این گرامر «id»، «num»، «create»، «delete»، «list»، «show»، «into»، «entry»، «insert» و «count» هستند.

```
S → create list L
S → delete list L
S → insert into L entry E
S → show list L
L → id
E → id count num
```

۴ (۱۵) گرامر روبرو را در نظر بگیرید. نشان دهید این گرامر ابهام دارد. گرامر را به صورتی تغییر دهید که عمل جمع شرکت‌پذیر چپ (Left-associative) باشد.

```
S → S + S
S → id ( S )
S → id
```

۵ (۳۵) برای گرامر سؤال قبل (بدون تغییرات خواسته شده در آن سؤال) جدول (`SLR(1)`) را محاسبه نمایید. آیا این گرامر (`SLR(1)`) است؟ همین کار را برای الگوریتم (`LALR(1)`) تکرار نمایید. جدول حاصل را به صورتی تغییر دهید که عملگر جمع شرکت‌پذیر چپ (Left-associative) باشد و با استفاده از آن رشته‌ی «`id + id(id + id)`» را تجزیه نمایید.