

اطلاعات کلی

■ ارائه دهنده: دکتر علی غلامی رودی (gholamirudi@nit.ac.ir)

■ کلاس: شنبه و دوشنبه کلاس ۲۰۸ از ۱۰:۰۰ تا ۱۱:۳۰

■ سایت درس: <http://nit.rudi.ir/>

معرفی

■ در درس طراحی کامپایلر، مفاهیم مربوط به کامپایلرها، معماری آنها و الگوریتم‌هایی که در پیاده‌سازی آنها استفاده می‌شوند، معرفی می‌گردند.

ارزشیابی

■ ارزشیابی در این درس با توجه به دو آزمون اصلی، سه تمرین کاغذی و تمرین برنامه‌نویسی انجام می‌شود.

■ در تمرین برنامه‌نویسی، کامپایلر زبان ساده‌ای به نام تسلنگ تا تولید کد میانی در سه گام پیاده‌سازی می‌شود.

منبع اصلی

■ A. W. Appel, Modern Compiler Implementation in C, Cambridge University Press, 1998.

منابع دیگر

■ K. D. Cooper, L. Torczon, Engineering a Compiler, Second Edition, Morgan Kaufmann, 2012.

■ A. V. Aho, M. S. Lam, R. Sethi, J. D. Ullman, Compilers: Principles, Techniques, and Tools, Second Edition, Addison Wesley, 2007.

برنامه‌ی نیمسال

موضوع	تاریخ
۱۳۹۸/۰۶/۲۳	مقدمه، اهمیت کامپایلر، کاربرد، نیازهای جدید، مفسرها، مترجم‌های زبان‌های مختلف
۱۳۹۸/۰۶/۲۵	ساختار کلی، معماری سه فازه، کد میانی، معرفی گام‌های هر فاز، ارتباط با سایر برنامه‌ها مثل لینکر
۱۳۹۸/۰۶/۳۰	تحلیل لغوی، تحلیل‌گر دست‌نویس، عبارت‌های منظم، بیان توکن‌ها با عبارت‌های منظم
۱۳۹۸/۰۷/۰۱	الگوریتم Thompson، الگوریتم ساختن زیر مجموعه‌ها، الگوریتم Hopcroft
۱۳۹۸/۰۷/۰۶	ترکیب عبارت‌های منظم، جستجو بدون ساختن DFA، بازیابی از خطا
۱۳۹۸/۰۷/۰۸	تحلیل نحوی، گرامرهای مستقل از متن، روش‌های تجزیه، در مورد گام اول تمرین عملی
۱۳۹۸/۰۷/۱۳	ابهام گرامر، تعیین اولویت و شرکت‌پذیری عملگرها با تغییر گرامر، ابهام if-else
۱۳۹۸/۰۷/۱۵	الگوریتم Recursive-Descent، تکنیک‌های رایج
۱۳۹۸/۰۷/۲۰	الگوریتم LL(1)، حذف چپ‌گردی، فاکتورگیری از چپ، معرفی LL(k)
۱۳۹۸/۰۷/۲۲	الگوریتم‌های تجزیه‌ی پایین به بالا، الگوریتم LR(0)
۱۳۹۸/۰۷/۲۷	—
۱۳۹۸/۰۷/۲۹	الگوریتم SLR، الگوریتم LR(1)
۱۳۹۸/۰۸/۰۴	LALR(1)، تعیین اولویت و شرکت‌پذیری عملگرها بدون تغییر گرامر
۱۳۹۸/۰۸/۰۶	روش‌های تصحیح خطا، تصحیح خطا به صورت محلی و سراسری، تصحیح خطا با توکن Error
۱۳۹۸/۰۸/۱۱	مقایسه‌ی قدرت الگوریتم‌های تجزیه، مطالب باقی‌مانده در مورد تجزیه، در مورد گام دوم تمرین عملی
۱۳۹۸/۰۸/۱۳	تحلیل مفهومی، کنش‌ها و مقادیر مفهومی، استفاده برای ارزیابی عبارت
۱۳۹۸/۰۸/۱۸	درخت مجرد، ساخت درخت مجرد با عملیات مفهومی
۱۳۹۸/۰۸/۲۰	جدول نمادها، استخراج نوع عبارت‌ها با کنش‌های مفهومی، بررسی نوع داده‌ها، تبدیل نوع‌ها
۱۳۹۸/۰۸/۲۵	کد میانی، انواع کد میانی، کد گرافی، کد خطی، کد سه آدرسه، کد میانی تسلیگ
۱۳۹۸/۰۸/۲۷	کدهای میانی SSA، تولید کد میانی با استفاده از کنش‌های مفهومی، در مورد گام سوم تمرین عملی
۱۳۹۸/۰۹/۰۲	امتحان میانترم
۱۳۹۸/۰۹/۰۴	مدیریت ایستا و پویای حافظه، سازماندهی پشته، فعال‌سازی روبه‌ها
۱۳۹۸/۰۹/۰۹	جمع‌آوری زباله، شیوه‌ی ارزیابی، روش شمارش مرجع، محدودیت‌ها، اصول روش‌های Trace-based
۱۳۹۸/۰۹/۱۱	تحلیل کد میانی، بلوک‌های پایه، گراف جریان
۱۳۹۸/۰۹/۱۶	تحلیل‌های جریان داده، زنده بودن متغیرها، تحلیل زنده بودن
۱۳۹۸/۰۹/۱۸	بهبودسازی‌های سراسری و محلی، انتشار ثوابت، تشخیص عبارت‌های مشابه
۱۳۹۸/۰۹/۲۳	گراف تداخل رجیسترها، تخصیص رجیستر با رنگ‌آمیزی گراف
۱۳۹۸/۰۹/۲۵	تولید کد نهایی، تفاوت‌های CISC و RISC برای تولید کد
۱۳۹۸/۰۹/۳۰	انتخاب دستورات
۱۳۹۸/۱۰/۰۲	زمانبندی دستورات، بهبودسازی‌های Peephole
۱۳۹۸/۱۰/۰۷	مباحثی از چالش‌های کامپایلرهای امروزی
۱۳۹۸/۱۰/۰۹	مطالب باقی‌مانده و مرور

موضوعات درس

۲

ساختار کامپایلر

اهمیت، کاربردها، نیازهای جدید، معماری سه فازه، گام‌های هر فاز.

۳

تحلیل لغوی

عبارات منظم، الگوریتم Thompson، الگوریتم ساختن زیرمجموعه‌ها، الگوریتم Hopcroft، ترکیب عبارتهای منظم، بدون ساختن DFA، تحلیلگر دست‌نویس.

۸

تحلیل نحوی

الگوریتم‌های بالا به پایین Recursive Descent، LL(1)، LL(2)، الگوریتم‌های پایین به بالای LR(0)، SLR، LR(1)، LALR(1)، ابهام گرامر، تعیین اولویت عملگرها با و بدون تغییر گرامر، روش‌های تصحیح خطا، تصحیح خطا با توکن Error.

۳

تحلیل مفهومی

عمل‌ها (کنش‌ها) و مقادیر مفهومی، استفاده برای ارزیابی نتیجه‌ی عبارت‌ها، درخت مجرد، جزئیات جدول نمادها، بررسی نوع داده، تبدیل نوع‌ها.

۲

کد میانی

کد سه-آدرسه، کد میانی زبان تسلینگ، کدهای میانی SSA، گراف‌های کد میانی مثل DAG، استفاده از کنش‌های مفهومی برای تولید کد میانی.

۳

محیط زمان اجرا و مدیریت حافظه

مدیریت ایستا و پویای حافظه، سازماندهی پشته، فعال‌سازی رویه‌ها و درخت‌های فعال‌سازی، جمع‌آوری زباله، روش «Reference counting» و محدودیت‌ها، اصول روش‌های Trace-based.

۶

تحلیل کد میانی

بلوک‌های پایه (Basic Blocks)، تحلیل زنده‌بودن متغیرها، گراف جریان، زمانبندی دستورات، بهینه‌سازی‌های محلی و سراسری، آشنایی با حذف کد مرده، انتشار ثوابت، تشخیص عبارت‌های مشابه.

۲

تخصیص رجیسترها

گراف تداخل رجیستر، تخصیص رجیستر سراسری با رنگ‌آمیزی گراف.

۲

تولید کد نهایی

تفاوت‌های معماری‌های RISC و CISC برای تولید کد، انتخاب دستور (Instruction selection)، بهینه‌سازی‌های Peephole.