

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

نظریه زبان‌ها و ماشین‌ها

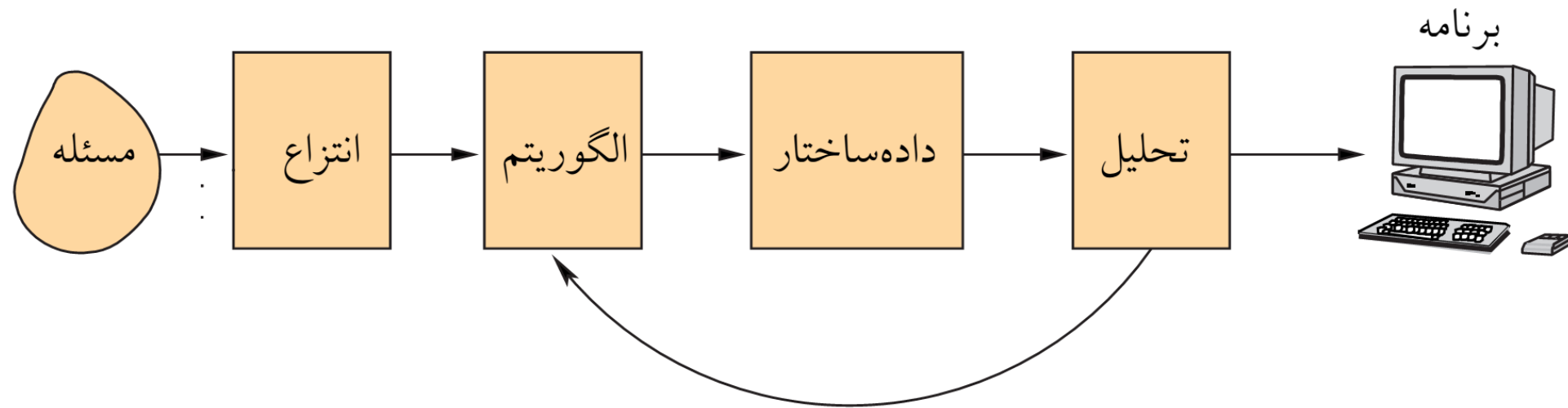
جلسه ۲

مجتبی خلیلی
دانشکده برق و کامپیوتر
دانشگاه صنعتی اصفهان

حضور و غیاب

○ بعد از ترمیم، حضور و غیاب خواهیم داشت.

مراحل حل مسئله



سه سوال اساسی در این درس

○ کامپیوترها چه مسائلی را می توانند حل کنند؟

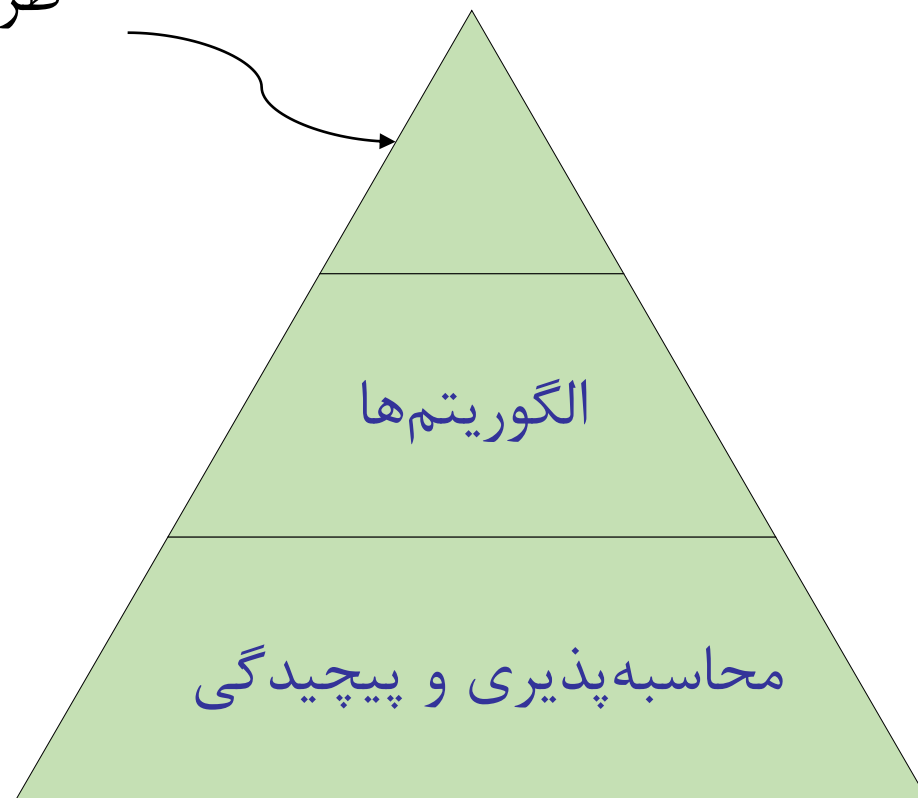
○ چه منابعی برای حل یک مسئله نیاز است؟

○ آیا برخی مسائل از برخی دیگر سختترند؟

❖ حل مسئله: بر مبنای ورودی، تصمیمی گرفته شود یا مقداری محاسبه شود.

حل مسئله

طراحی و پیاده‌سازی
نرم‌افزاری



الگوریتم

○ قلب برنامه‌های کامپیوتری، الگوریتم‌ها هستند.

○ برای مطالعه الگوریتم‌ها ما باید بتوانیم به صورت ریاضی درباره موارد زیر صحبت کنیم:

- مسائل محاسباتی
- کامپیوترها
- الگوریتم‌ها

درباره الگوریتم‌ها

○ آیا درستند؟

درس ساختمان گسسته

○ آیا روش بهتری برای حل همان مسئله وجود دارد؟

درس طراحی الگوریتم-نظریه پیچیدگی

○ آیا هر مسئله جواب دارد؟

نظریه محاسبه پذیری

○ اگر دارد، آیا این جواب را می‌توان با یک الگوریتم بدست آورد؟

آیا هر مسئله یک الگوریتم دارد؟

مسئله

○ چند مثال:

- تحت n عدد صحیح، لیست مرتب شده آن را بیابید.
- در یک گراف، کوتاهترین مسیر بین دو گره مشخص را بیابید.
- تجزیه $n=pq$ به فاکتورهایش را بیابید.
- تعیین کنید آیا یک چند جمله‌ای چند متغیره با ضرایب صحیح دارای جواب صحیح است یا خیر.

درباره الگوریتم

- ما برای همه مسائل الگوریتم نداریم (یک اثبات برای آن وجود دارد).
- برای بسیاری از مسائل، متاسفانه یا خوشبختانه ما اطلاعات کمی درباره سریع‌ترین الگوریتم‌های متناظر آنها داریم.
- مثل فاکتور کردن $n=pq$
- چنانچه الگوریتم سریعی برای آنها پیدا شود، دنیا تغییر خواهد کرد.

کامپیوترها می توانند ...

امکان ناپذیری

- چرا علاقه داریم درباره امکان ناپذیری برخی مسائل تحقیق کنیم؟
- مثلاً، زمانی تصور میشد میتوان ماشینی ساخت که انرژی مصرف نکند.
- بعدها، فیزیکدانان نشان دادند چنین چیزی امکان ندارد.

فهم امکان ناپذیرها کمک می کند وقت صرف آنها نکنید.

مثال

○ تعیین کنید آیا یک چند جمله‌ای چند متغیره با ضرایب صحیح دارای جواب صحیح است یا خیر.

مثال

○ مسئله رنگ آمیزی گراف: تخصیص رنگ به رئوس های یک گراف به طوری که هیچ دو رئوس مجاور هم رنگ نباشند و تعداد رنگ ها کمینه باشد.

مثال

○ جنبه دیگر آن برای توسعه دهندگان نرم افزار این است که تشخیص دهند یک مسئله قابل حل (به صورت عام یا در زمان کارآمد) است یا خیر.

قوانین محاسبه

- همانگونه که قوانین فیزیک درباره ممکن و ناممکن در طبیعت صحبت میکنند، برای محاسبات نیز قوانینی داریم که درباره ممکن و ناممکن برای یک کامپیوتر صحبت می کنند.
- این قوانین، اغلب در بحث نظریه محاسبه مطرح می شوند.

نظریه محاسبه

○ تعریف محاسبه در این درس: پردازش اطلاعات با بکارگیری نامحدود یکسری عملیات یا قواعد معلوم و متناهی

○ منظور از نظریه این است که ایده کلی را به صورت انتزاعی، دقیق و ساده بیان کنیم.

- مستقل از تکنولوژی
- حذف موارد غیرضروری
- قابل اثبات با ابزارهای ریاضی

○ مثال

نظریه محاسبه

○ نظریه محاسبه: نظریه‌ای ریاضی درباره مدل‌های اساسی محاسبه و بحث درباره چگونگی نمایش، توانایی‌ها و محدودیت‌های این مدل‌ها

نظریه محاسبه

- مدل کردن ماشین‌ها (دستگاه)
- مدل کردن مسائلی که بوسیله ماشین می‌توان حل کرد.
- قضایایی که بیان کنند چه ماشین‌هایی می‌توانند چه مسائلی را با چه هزینه‌ای حل کنند.

برای این درس، مسائل ما اغلب به این صورت است که تشخیص دهیم آیا یک رشته متعلق به یک زبان است یا خیر؟

نظریه محاسبه

○ این نظریه معمولاً به سه دسته زیر تقسیم می‌شود:

- نظریه اتوماتا
- نظریه محاسبه پذیری
- نظریه پیچیدگی

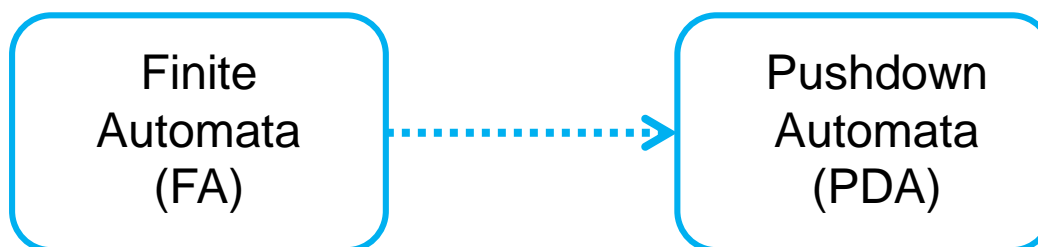
سه مدل

Finite
Automata
(FA)

○ اتوماتای متناهی

- حافظه محدود
- زبان‌ها، عبارات و گرامرهای منظم
- جستجو در متن

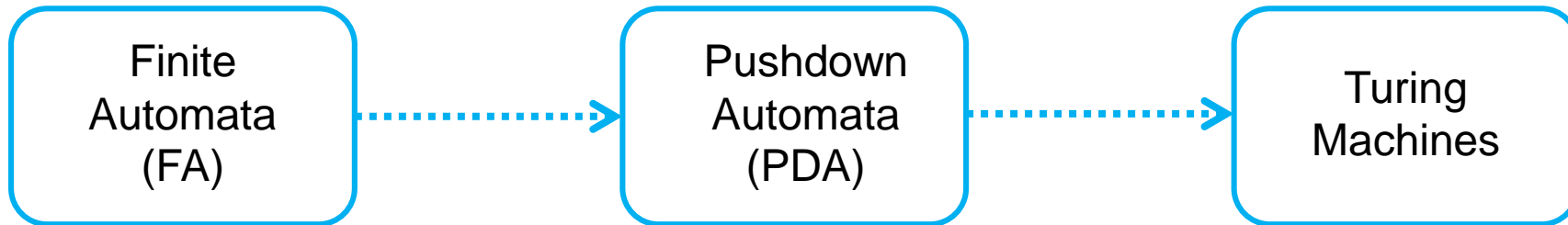
سه مدل



○ اتوماتای پشته‌ای

- حافظه نامحدود اما دسترسی خاص (stack)
- زبان‌ها و گرامرهای مستقل از متن
- زبان‌های برنامه نویسی، کامپایلر

سه مدل



○ ماشین تورینگ

- حافظه نامحدود
- زبان‌های حساس به متن
- کامپیوترها
- محاسبه پذیری، پیچیدگی

جمع بندی

○ حل مسئله‌ها

