## Lecture 13: Parsing & Logic

CSCI 81 Spring, 2015

Kim Bruce

## Arithmetic grammar

Look at parse tree & abstract syntax tree for 2 \* 3 + 7

#### Recursive Descent Parser

```
Base recognizer (ignore building tree) on productions:
```

#### **Problems**

- How do we select which production to use when alternatives?
- Left-recursive never terminates

#### Rewrite Grammar

#### Rewrite Grammar

```
<exp> ::= <term> <termTail>
                                                      (1)
 <termTail> ::= <addop> <term> <termTail>
                                                      (2)
                                                     (3)
      <term> ::= <factor> <factorTail>
                                                      (4)
<factorTail> ::= <mulop> <factor> <factorTail>
                                                      (5)
                                                     (6)
    <factor> ::= ( <exp> )
                                                      (7)
                 NUM
                                                      (8)
                 ID
                                                      (9)
     <addop> ::= + | -
                                                     (10)
     <mulop> ::= * | /
                                                     (11)
                     No left recursion
         How do we know which production to take?
```

## Predictive Parsing (LL(1))

 $Goal: a_{\scriptscriptstyle \rm I}a_{\scriptscriptstyle \rm 2}...a_n$ 

$$S \rightarrow \alpha$$
...
 $\rightarrow a_1 a_2 X \beta$ 

Want next terminal character derived to be a<sub>3</sub>

 $a_3$  in First( $\gamma$ )

Need to apply a production  $X := \gamma$  where

- 1)  $\gamma$  can eventually derive a string starting with  $a_3$  or
- 2) If X can derive the empty string, then see if  $\beta$  can derive a string starting with  $a_3$ .

 $a_3$  in Follow(X)

Non- terminals	ID	NUM	Addop	Mulop	(	)	EOF
< <i>exp&gt;</i>	I	I			I		
<termtail></termtail>			2			3	3
<term></term>	4	4			4		
<facttail &gt;</facttail 			6	5		6	6
<factor></factor>	9	8			7		
<addop></addop>			IO				
<mulop></mulop>				II			

Read off from table which production to apply!  $Ex: Parse \ 2 * 3 + 7$ 

## Logic(s)

## Logic

- Context free language designed for expressing Boolean-valued statements
- Goal is to investigate when logic statements are
  - True in some or all models
  - Provable according to rules for proving
  - ... and to see if there is a connection between the two
- Start simple & work up in complexity.

## Propositional Logic

- Definition of well-formed formulas of prop logic:
  - $S ::= P | (S \lor S) | (S \land S) | (\neg S) | (S \rightarrow S)$

Use "::=" in place of "→"

• P := p | q | r | ...

for productions to avoid confusion

- Often (informally) drop parentheses around terms
  - Precedence:  $\neg$ ,  $\wedge$ ,  $\vee$ ,  $\rightarrow$
  - ∧ and v are left associative; → is right associative.
- Sometimes add  $\top$  for true and  $\bot$  for false.

## Semantics of Propositional Logic

- Meaning of formula depends on meaning of propositional letters.
  - Start with valuation fcn V: Prop Letters → {true,false}
  - Extend to V\*: Prop Logic Formulas  $\rightarrow$  {true, false} by
    - $V^*(p) = V(p)$  if p is propositional letter
    - $V^*(\neg \varphi)$  = false iff  $V^*(\varphi)$  = true
    - $V^*(\phi \vee \gamma)$  = true iff  $V^*(\phi)$  = true or  $V^*(\gamma)$  = true (or both)
    - $V^*(\phi \wedge \gamma)$  = true iff  $V^*(\phi)$  = true and  $V^*(\gamma)$  = true
    - $V^*(\phi \rightarrow \gamma)$  = false iff  $V^*(\phi)$  = true and  $V^*(\gamma)$  = false

#### **Truth Tables**

P	Q	$\neg P$	$P \wedge Q$	Pv $Q$	<i>P</i> →2
T	T	F	Т	Т	Т
T	F	F	F	Т	F
F	T	Т	F	Т	Т
F	F	Т	F	F	Т

Each row corresponds to different valuation

## Categories of WFFs

- A formula  $\phi$  is *valid*, or a *tautology*, if for all valuations V, we have  $V^*(\phi) = \text{true}$ .
- A formula  $\phi$  is *satisfiable* if for some valuation V, we have V\*( $\phi$ ) = true.
- A formula  $\phi$  is *falsifiable* if for some valuation V, we have V\*( $\phi$ ) = false.
- A formula  $\phi$  is *unsatisfiable*, or a *contradiction*, if for all valuations V, we have  $V^*(\phi)$  = false.

#### Semantic Entailment

- $\phi_{r}$ , ...,  $\phi_{n} \models \psi$  iff for every valuation V s.t.  $V^{*}(\phi_{r}) = ... = V^{*}(\phi_{n}) = true$ , then  $V^{*}(\psi) = true$ 
  - Example:  $P \models Q \rightarrow P$
  - Read  $\phi_i$ , ...,  $\phi_n \models \psi$  as  $\phi_i$ , ...,  $\phi_n$  semantically entails  $\psi$
- Hence,  $\models \psi$  iff  $\psi$  is a tautology.
- Show:  $\phi_1, ..., \phi_n, \phi \models \psi \text{ iff } \phi_1, ..., \phi_n \models \phi \rightarrow \psi$

#### **Proof Rules**

- Syntactically determined set of rules for inferring conclusion from hypotheses.
- Rules provide kind of meaning for connectives
  - Different texts use different rules -- all equivalent!
- $\phi_1, ..., \phi_n \vdash \psi$
- Constructing proof is creative
  - Not clear what rules to apply

## Rules for A

$$\frac{\varphi \quad \psi}{\varphi \quad \wedge \psi} \wedge i$$

$$\frac{\phi \wedge \psi}{\phi} \wedge e_{r}$$

$$\frac{\phi \wedge \psi}{\psi} \wedge e_2$$

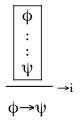
## Rules for $\neg$ , $\rightarrow$

$$\frac{\phi \quad \phi \rightarrow \psi}{} \rightarrow e$$

modus ponens

modus tolens
\*derived rules

## → Introduction

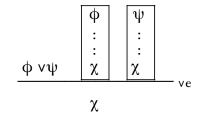


If from an assumption of  $\phi$ , one can deduce  $\psi$ , then one can deduce  $\phi \rightarrow \psi$ I.e., hypothesis of  $\phi$  is discharged in proof.

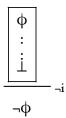
## v Rules

$$\frac{\Phi}{\Phi \vee \psi}$$
 via

$$\frac{\psi}{\varphi \vee \psi} \vee i_2$$



## Negation & ⊥-Rules



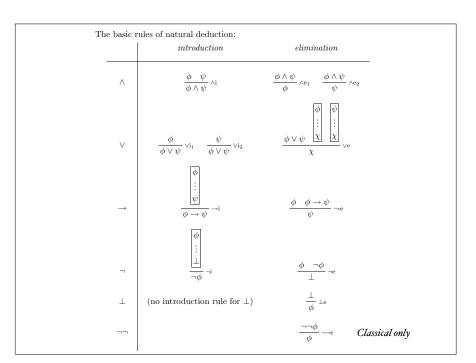
## Proofs

- Ordered list of steps where each step justified as premise or by proof rule from earlier steps.
- Show  $\vdash \neg(\varphi \land \neg \varphi)$

1. ф∧¬ф	assumption
2. ф	ı, ۸e
3. ¬ф	ı, ۸e
4. ⊥	2,3, ¬e
5. ¬(φ ∧ ¬φ)	) <sub>1-4</sub> , ¬i

Always indicate proof rule and steps used to get new wff Use boxes for subproofs to be discharged

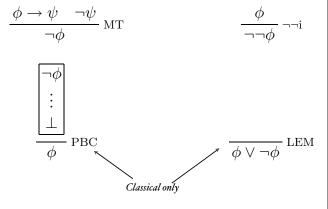
Distinction between hypothesis and assumption



## **Example Proofs**

- Be careful with proof boxes:
  - Can't use internal steps when reasoning outside the box.
- Typically work backwards!
- Show
  - $\phi \lor \psi \vdash \psi \lor \phi$
  - $\vdash \varphi \rightarrow (\psi \rightarrow \varphi)$
  - $\phi \rightarrow (\psi \rightarrow \chi) \vdash \psi \rightarrow (\phi \rightarrow \chi)$

Some useful derived rules:



### Troublesome Proof

- Are there two irrational numbers, *a* and *b*, such that *a*<sup>*b*</sup> is rational?
  - Notice  $(\sqrt{2}^{\sqrt{2}})^{\sqrt{2}} = \sqrt{2^2} = 2$
  - Case 1:  $\sqrt{2^{1/2}}$  is rational and take  $a = b = \sqrt{2}$
  - Case 2:  $\sqrt{2^{\sqrt{2}}}$  is irrational and take  $a = \sqrt{2^{\sqrt{2}}}$ ,  $b = \sqrt{2}$
- Constructivist rejects because can't tell which alternative is true.

# Constructivist vs Classical Logics

- Constructivists don't believe in ¬¬e rule:
  - ¬¬ф ⊢ф
- Don't believe  $\vdash \varphi \lor \neg \varphi$  except in special cases.
  - Don't accept proof by contradiction!
- Give constructive proof of  $\phi \lor \psi \vdash \neg(\neg \phi \land \neg \psi)$