Predicate Calculus
- Semantic Tableau (2/2)

Moonzoo Kim
CS Division of EECS Dept.
KAIST

Formal construction

- Formal construction is explained in two steps
  1. Construction rules (α rule, β rule, γ rule for $\forall x$, and δ rule for $\exists y$)
     - These rules might not be systematic, but enough for showing soundness of a semantic tableau.

$\forall$x

γ	$\gamma(a)$
$\forall x A(x)$	$A(a)$
$\neg\exists x A(x)$	$\neg A(a)$

$\exists$x

δ	$\delta(a)$
$\exists x A(x)$	$A(a)$
$\neg\forall x A(x)$	$\neg A(a)$

  2. Systematic construction rules, which specify the order of applying rules
     - Systematic construction rules can show the completeness of a semantic tableau

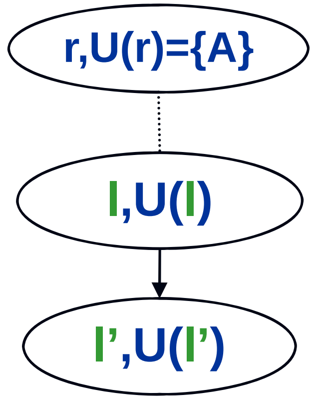
- Def 5.25 A literal is a closed atomic formula $p(a_1,\ldots,a_k)$ or the negation of such a formula
  - If a formula has no free variable, it is closed. Therefore, if an atomic formula is closed, all of its arguments are constants.

Formal construction rules (1/2)

- Alg 5.26 (Construction of a semantic tableau)
  - Input: A formula A of the predicate calculus
  - Output: A semantic tableau $\mathcal{T}$ for A
    - Each node of $\mathcal{T}$ will be labeled with a set of formulas.
      - Initially, $\mathcal{T}$ consists of a single node, the root, labeled with {A}
    - All branches are either
      - infinite or
      - finite with
        - leaves marked closed or
        - leaves marked open
    - $\mathcal{T}$ is built inductively by choosing an unmarked leaf l labeled with a set of formulas U(l), and applying one of the following rules:

Formal construction rules (2/2)

- If U(l) is a set of literals and γ–formulas which contains a complementary pair of literals $\{p(a_1,\ldots,a_k), \neg p(a_1,\ldots,a_k)\}$, mark the leaf closed x
- If U(l) is not a set of literals, ***choose*** a formula A in U(l) which is not a literal
  - if A is an α-formula or β-formula, do the same as in propositional calculus
  - if A is a γ–formula (such as $\forall x\, A_1(x)$), create a new node l' as a child of l and label l' with U(l') = U(l) ∪ {γ(a)} where a is some constant that appears in U(l) (**infinite branch)**
    - If no constant exists in U(l), use an arbitrary constant, say a_i
    - Note that the γ-formula remains in U(l').
    - If U(l) consists only of literals and γ-formulas and U(l) does not contain a complementary pair of literals and U(l')=U(l) for all choices of a, then mark the leaf as open O. **(finite branch)**
      - If the only rule that applies is a γ–rule and the rule produces no new subformulas, then the branch is open.
        - ex. for $\{\forall x\, p(a,x)\}$, ({a},{(a,a)},{a}) is a model for it.
  - if A is a δ formula (such as $\exists x\, A_1(x)$), create a new node l' as a child of l and label l' with U(l') = (U(l) – {A}) ∪ {δ(a)} where a is some constant that does not appear in U(l).



Soundness

- Thm 5.28 (Soundness) let A be a formula in the predicate calculus and let $\mathcal{T}$ be a tableau for A. If $\mathcal{T}$ closes, then A is unsatisfiable.
  - However, the construction of the tableau is not complete unless it is built systematically.
    - ex. $\forall x \exists y\, p(x,y) \wedge \forall x (p(x) \wedge \neg p(x))$
- The proof is by induction on the height h of node n
  - Cases for h=0, and the inductive cases for α, β formulas is the same as the proof in the propositional calculus
  - Case 3: The γ–rule was used. Then
    - $U(n) = U_0 \cup \{\forall x\, A(x)\}$ and $U(n') = U_0 \cup \{\forall x\, A(x), A(a)\}$
    - Assume that U(n) is satisfiable and let $\mathcal{I}$ be a model for U(n), so that $v_{\mathcal{I}}(A_i) = T$ for all $A_i \in U_0(n)$ and also $v_{\mathcal{I}}(\forall x\, A(X)) = T$.
    - By Thm 5.15, $v_{\mathcal{I}}(\forall x\, A(x)) = T$ iff $v_{\sigma_{\mathcal{I}}} = T$ for all assignments $\sigma_{\mathcal{I}}$, in particular for any assignment that assigns the same domain element to x that $\mathcal{I}$ does to a
    - But $v_{\mathcal{I}}(A(a)) = T$ contradicts the inductive hypothesis that U(n') is unsatisfiable
  - Case 4: The δ-rule was used. Then
    - $U(n) = U_o \cup \{\exists x\, A(x)\}$ and $U(n') = U_0 \cup \{A(a)\}$ for some constant a which does not occur in a formula of U(n)
    - Assume that U(n) is satisfiable and let $\mathcal{I} = (D, \{R_1, \cdots, R_n\}, \{d_1, \cdots, d_k\})$ be a satisfying interpretation.
    - Then $v_{\mathcal{I}}(\exists x A(x)) = T$, so for the relation R_i assignmed to A and for some $d \in D$, $(d) \in R_i$. Extend $\mathcal{I}$ to the interpretation $\mathcal{I}' = (D, \{R_1, \cdots, R_n\}, \{d_1, \cdots, d_k, d\})$ by assigning d to the constant a.
    - Then $v_{\mathcal{I}'}(A(a)) = T$, and since $v_{\mathcal{I}'}(U_0) = v_{\mathcal{I}}(U_0) = T$, we can conclude that $v_{\mathcal{I}'}(U(n')) = T$, contradicting the inductive hypothesis that U(n') is unsatisfiable

Systematic formal construction rules (1/2)

- The aim of the systematic construction is to ensure that
  1. rules are eventually applied to all formulas in the label of a node and
  2. in the case of universally quantified formulas, that an instance is created for all constants that appears
- Alg 5.29 (Systematic construction of a semantic tableau)
  - Input: A formula A of the predicate calculus
  - Output: A semantic tableau $\mathcal{T}$ for A
    - key idea: to apply α,β,δ, and γ rules in order, to prevent infinite branch due to γ rule from hidding that an branch is closed
  - A semantic tableau for A is a tree $\mathcal{T}$ each node of which is labeled by a pair W(n) = (U(n),C(n)), where U(n) = $\{A_1,\ldots,A_k\}$ is a set of formulas and C(n) = $\{a_1,\ldots,a_m\}$ is a set of constants appearing in the formulas in U(n)
  - Initially, $\mathcal{T}$ consists of a single node (the root) labeled with $(\{A\},\{a_1,\ldots,a_m\})$
    - If A has no constants, choose an arbitrary constant a and label the node with ({A},{a})

Systematic formal construction rules (2/2)

- Inductively applying ofe of the following rules in the order given
  1. If U(l) is a set of literals and γ–formulas which contains a complementary pair of literals $\{p(a_1,\ldots,a_k), \neg p(a_1,\ldots,a_k)\}$, mark the leaf closed x
  2. If U(l) is not a set of literals, ***choose*** a formula A in U(l) which is not a literal
     1. if A is an α-formula or β-formula, do the same as in propositional calculus with C(l')=C(l)
     2. if A is a δ–formula, create a new node l' as a child of l and label l' with W(l') = $((U(l)-\{A\})\cup \{\delta(a)\}, C(l)\cup\{a\})$ where a is some constant that does not appears in U(l)
  3. Let $\{\gamma_1,\ldots,\gamma_m\} \subseteq U(l)$ be all the γ–formulas in U(l) and let C(l) $=\{a_1,\ldots,a_k\}$. Create a new node l' as a child of l and label l' with
     - $W(l') = (U(l) \cup \bigcup_{i=1..m,j=1..k}\{\gamma_i(a_j)\},C(l)\}$
     - If U(l) consists only of literals and γ–formulas and U(l) does not contain a complementary pair of literals and U(l') = U(l), then mark the leaf as open O.

Completeness

- Thm 5.34 (Completeness) Let A be a valid formula. Then the systematic semantic tableau for ¬A closes
  - Thm 5.32 Let b be an open branch of a systematic tableau and U = $\bigcup_{n \in b}$ U(n). The U is a Hintikka set.
  - Lem 5.33 (Hintikka's lemma) Let U be a Hintikka set. Then there is a model for U
- Proof:
  - Let A be a valid formula and suppose that the systematic tableau for ¬A does not close.
  - By Thm 5.32, there is an open branch b s.t. U = $\bigcup_{n \in b}$ U(n) is a Hintikka set.
  - By Lem 5.33, there is a model $\mathcal{I}$ for U. But ¬A ∈ U so $\mathcal{I} \models \neg A$ contradicting the assumption that A is valid