CS 228 : Logic in Computer Science

Krishna. S

Recap

- Signature τ: set containing relations and constants

Recap

- Signature τ: set containing relations and constants
- Each relation in τ has an arity

Recap

- Signature τ: set containing relations and constants
- Each relation in τ has an arity
- A FO formula is written over some signature τ; that is, it uses the relations and constants from τ. It also uses variables denoted x_i, boolean connectives and quantifiers $\forall, \exists$.

Recap

- Signature τ: set containing relations and constants
- Each relation in τ has an arity
- A FO formula is written over some signature τ; that is, it uses the relations and constants from τ. It also uses variables denoted x_i, boolean connectives and quantifiers $\forall, \exists$.
- A relation R of arity k is used in the formula as $R(t_1, \ldots, t_k)$ where t_i's are variables or constants

Recap

- Signature τ: set containing relations and constants
- Each relation in τ has an arity
- A FO formula is written over some signature τ; that is, it uses the relations and constants from τ. It also uses variables denoted x_i, boolean connectives and quantifiers $\forall, \exists$.
- A relation R of arity k is used in the formula as $R(t_1, \ldots, t_k)$ where t_i's are variables or constants
- Equality $t_1 = t_2$ is available irrespective of τ

Recap

- Signature τ: set containing relations and constants
- Each relation in τ has an arity
- A FO formula is written over some signature τ; that is, it uses the relations and constants from τ. It also uses variables denoted x_i, boolean connectives and quantifiers $\forall, \exists$.
- A relation R of arity k is used in the formula as $R(t_1, \ldots, t_k)$ where t_i's are variables or constants
- Equality $t_1 = t_2$ is available irrespective of τ
- To make sense out of a formula, we need structures

Recap

- A structure of signature τ consists of a universe, and assigns meanings to all the entities of τ

Recap

- A structure of signature τ consists of a universe, and assigns meanings to all the entities of τ
- So, if R is a k-ary relational symbol of τ, the structure specifies which k-tuples of elements from the universe are legitimate relations for R
- If c is a constant in τ, the structure also maps c to some fixed element of the universe

Recap

- A structure of signature τ consists of a universe, and assigns meanings to all the entities of τ
- So, if R is a k-ary relational symbol of τ, the structure specifies which k-tuples of elements from the universe are legitimate relations for R
- If c is a constant in τ, the structure also maps c to some fixed element of the universe
- So, structures of τ give life to τ

Recap

- A structure of signature τ consists of a universe, and assigns meanings to all the entities of τ
- So, if R is a k-ary relational symbol of τ, the structure specifies which k-tuples of elements from the universe are legitimate relations for R
- If c is a constant in τ, the structure also maps c to some fixed element of the universe
- So, structures of τ give life to τ
- Structures also tell you the set of values your variables x_i can assume : these are the elements from the universe

Recap

- A structure of signature τ consists of a universe, and assigns meanings to all the entities of τ
- So, if *R* is a *k*-ary relational symbol of τ, the structure specifies which *k*-tuples of elements from the universe are legitimate relations for *R*
- If *c* is a constant in τ, the structure also maps *c* to some fixed element of the universe
- So, structures of τ give life to τ
- Structures also tell you the set of values your variables x_i can assume : these are the elements from the universe
- A structure in PL will just consist of the universe $\{0, 1\}$, since there is no signature. All variables assume values from this boolean universe.

Satisfiability in PL and FO

- The satisfiability of a PL formula depends on the existence of an assignment satisfying it; likewise, the satisfiability of a FO formula φ over signature τ depends on the existence of a structure $\mathcal{A}$ of τ such that φ is true on $\mathcal{A}$.

Free and Bound Variables

- For a wff $\varphi = \forall x \psi$, ψ is said to be the scope of the quantifier x

Free and Bound Variables

- For a wff $\varphi = \forall x \psi$, ψ is said to be the scope of the quantifier x
- Every occurrence of x in $\forall x \psi$ is bound
- Any occurrence of x which is not bound is called free

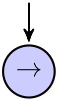
Free and Bound Variables

- For a wff $\varphi = \forall x \psi$, ψ is said to be the scope of the quantifier x
- Every occurrence of x in $\forall x \psi$ is bound
- Any occurrence of x which is not bound is called free
- $\varphi = P(x, y) \to \forall x((\forall y R(x, y)) \to Q(x, y))$
  - y is free in $Q(x, y)$ and bound in $R(x, y)$,
  - x is free in $P(x, y)$, and bound in $Q(x, y)$, $R(x, y)$

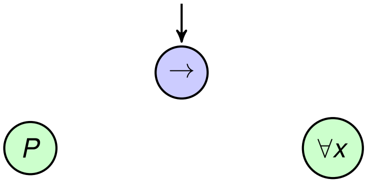
Free and Bound Variables

- For a wff $\varphi = \forall x \psi$, ψ is said to be the scope of the quantifier x
- Every occurrence of x in $\forall x \psi$ is bound
- Any occurrence of x which is not bound is called free
- $\varphi = P(x, y) \rightarrow \forall x((\forall y R(x, y)) \rightarrow Q(x, y))$
  - y is free in $Q(x, y)$ and bound in $R(x, y)$,
  - x is free in $P(x, y)$, and bound in $Q(x, y), R(x, y)$
- Given φ, denote by $\varphi(x_1, \ldots, x_n)$, that $x_1, \ldots, x_n$ are the free variables of φ, also *free*(φ)
- A sentence is a formula φ none of whose variables are free

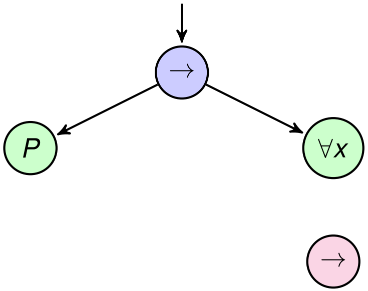
$$P(x, y) \to \forall x((\forall y R(x, y)) \to Q(x, y))$$



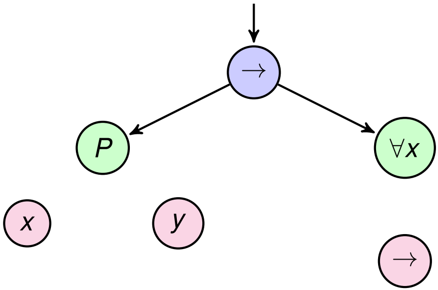
$$P(x, y) \to \forall x((\forall y R(x, y)) \to Q(x, y))$$



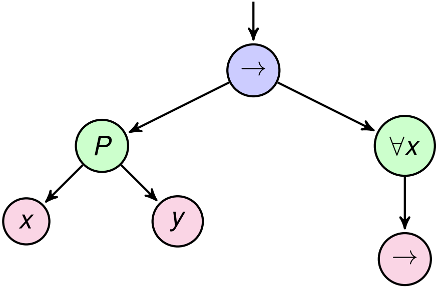
$P(x,y) \to \forall x((\forall y R(x,y)) \to Q(x,y))$



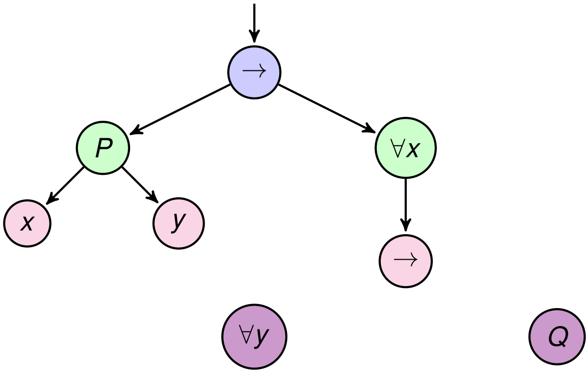
$P(x,y) \to \forall x((\forall y R(x,y)) \to Q(x,y))$



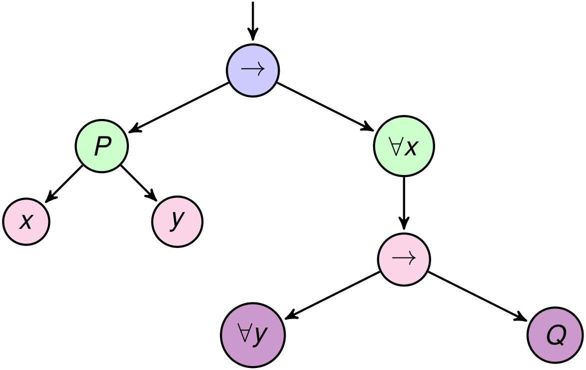
$P(x, y) \to \forall x((\forall y R(x, y)) \to Q(x, y))$



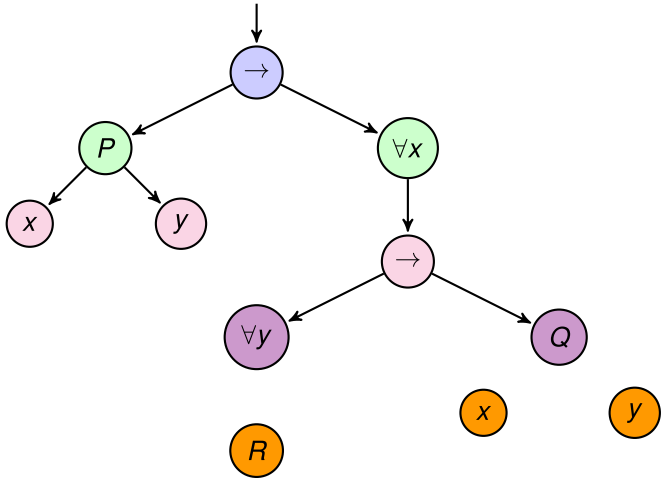
$P(x, y) \to \forall x((\forall y R(x, y)) \to Q(x, y))$



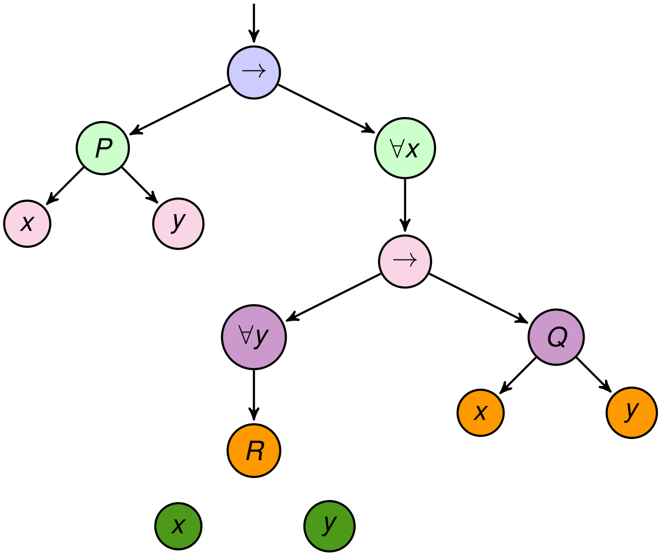
$P(x, y) \to \forall x((\forall y R(x, y)) \to Q(x, y))$



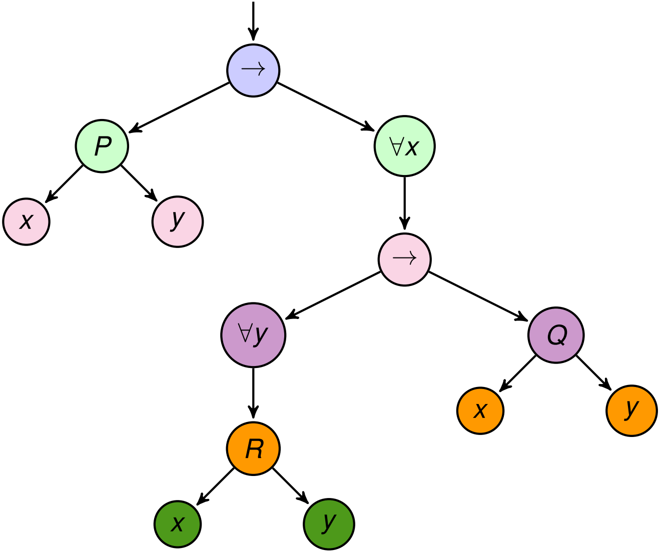
$$P(x,y) \rightarrow \forall x((\forall y R(x,y)) \rightarrow Q(x,y))$$



$P(x, y) \to \forall x((\forall y R(x, y)) \to Q(x, y))$

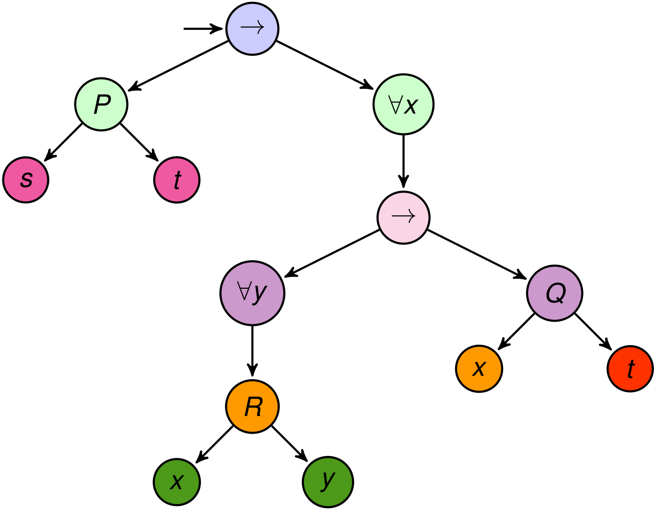


$P(x, y) \to \forall x((\forall y R(x, y)) \to Q(x, y))$

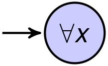


$P(x, y) \to \forall x((\forall y R(x, y)) \to Q(x, y))$

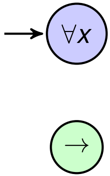
$\varphi(s, t) = P(s, t) \to \forall x((\forall y R(x, y)) \to Q(x, t))$



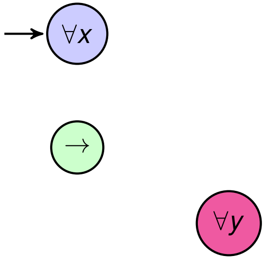
$\forall x(R(x, y) \to \forall y P(y))$



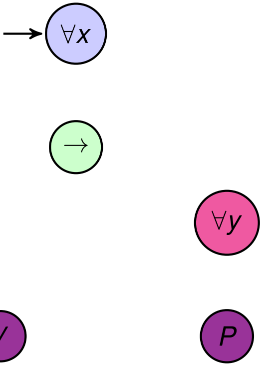
$\forall x(R(x, y) \to \forall y P(y))$



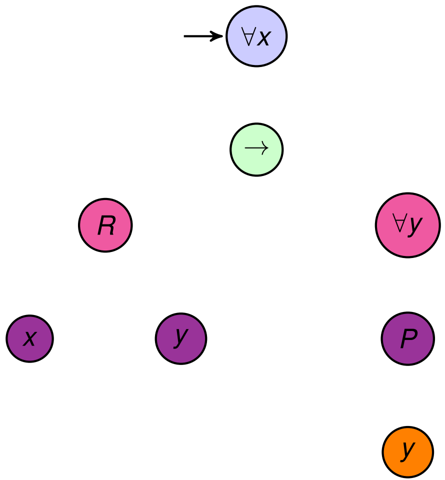
$\forall x(R(x,y) \to \forall y P(y))$



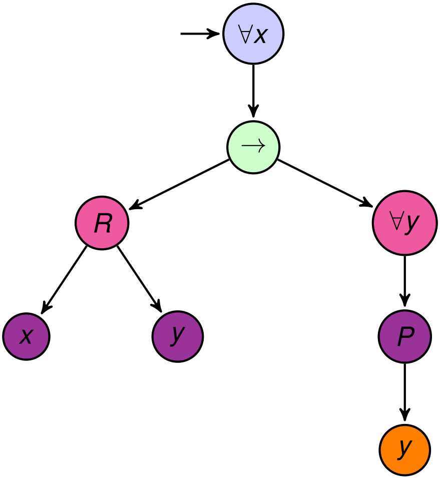
$\forall x(R(x,y) \to \forall y P(y))$



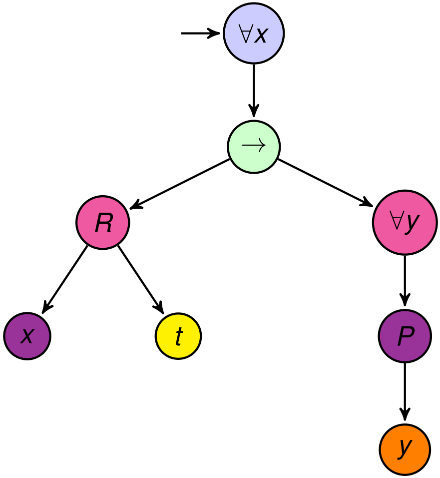
$\forall x(R(x,y) \to \forall y P(y))$



$\forall x(R(x,y) \to \forall y P(y))$



$\forall x(R(x,y) \to \forall y P(y))$



$\varphi(t) = \forall x(R(x,t) \to \forall y P(y))$

Assignments on τ-structures

Assignments

For a τ-structure $\mathcal{A}$, an assignment over $\mathcal{A}$ is a function $\alpha : \mathcal{V} \to u(\mathcal{A})$ that assigns every variable $x \in \mathcal{V}$ a value $\alpha(x) \in u(\mathcal{A})$. If t is a constant symbol c, then $\alpha(t)$ is $c^{\mathcal{A}}$

Assignments on τ-structures

Assignments

For a τ-structure $\mathcal{A}$, an assignment over $\mathcal{A}$ is a function $\alpha : \mathcal{V} \to u(\mathcal{A})$ that assigns every variable $x \in \mathcal{V}$ a value $\alpha(x) \in u(\mathcal{A})$. If t is a constant symbol c, then $\alpha(t)$ is $c^{\mathcal{A}}$

Binding on a Variable

For an assignment α over $\mathcal{A}$, $\alpha[x \mapsto a]$ is the assignment
$\alpha[x \mapsto a](y) = \begin{cases} \alpha(y), y \neq x, \\ a, y = x \end{cases}$

Satisfaction

We define the relation $\mathcal{A} \models_\alpha \varphi$ (read as φ is true in $\mathcal{A}$ under the assignment α) inductively:

Satisfaction

We define the relation $\mathcal{A} \models_\alpha \varphi$ (read as φ is true in $\mathcal{A}$ under the assignment α) inductively:

- $\mathcal{A} \not\models_\alpha \bot$

Satisfaction

We define the relation $\mathcal{A} \models_\alpha \varphi$ (read as φ is true in $\mathcal{A}$ under the assignment α) inductively:

- $\mathcal{A} \not\models_\alpha \bot$
- $\mathcal{A} \models_\alpha t_1 = t_2$ iff $\alpha(t_1) = \alpha(t_2)$

Satisfaction

We define the relation $\mathcal{A} \models_\alpha \varphi$ (read as φ is true in $\mathcal{A}$ under the assignment α) inductively:

- $\mathcal{A} \not\models_\alpha \bot$
- $\mathcal{A} \models_\alpha t_1 = t_2$ iff $\alpha(t_1) = \alpha(t_2)$
- $\mathcal{A} \models_\alpha R(t_1, \ldots, t_k)$ iff $(\alpha(t_1), \ldots, \alpha(t_k)) \in R^{\mathcal{A}}$

Satisfaction

We define the relation $\mathcal{A} \models_\alpha \varphi$ (read as φ is true in $\mathcal{A}$ under the assignment α) inductively:

- $\mathcal{A} \not\models_\alpha \bot$
- $\mathcal{A} \models_\alpha t_1 = t_2$ iff $\alpha(t_1) = \alpha(t_2)$
- $\mathcal{A} \models_\alpha R(t_1, \ldots, t_k)$ iff $(\alpha(t_1), \ldots, \alpha(t_k)) \in R^{\mathcal{A}}$
- $\mathcal{A} \models_\alpha (\varphi \to \psi)$ iff $\mathcal{A} \not\models_\alpha \varphi$ or $\mathcal{A} \models_\alpha \psi$

Satisfaction

We define the relation $\mathcal{A} \models_\alpha \varphi$ (read as φ is true in $\mathcal{A}$ under the assignment α) inductively:

- $\mathcal{A} \not\models_\alpha \bot$
- $\mathcal{A} \models_\alpha t_1 = t_2$ iff $\alpha(t_1) = \alpha(t_2)$
- $\mathcal{A} \models_\alpha R(t_1, \ldots, t_k)$ iff $(\alpha(t_1), \ldots, \alpha(t_k)) \in R^{\mathcal{A}}$
- $\mathcal{A} \models_\alpha (\varphi \to \psi)$ iff $\mathcal{A} \not\models_\alpha \varphi$ or $\mathcal{A} \models_\alpha \psi$
- $\mathcal{A} \models_\alpha (\forall x)\varphi$ iff for every $a \in u(\mathcal{A})$, $\mathcal{A} \models_{\alpha[x \mapsto a]} \varphi$

Satisfaction

We define the relation $\mathcal{A} \models_\alpha \varphi$ (read as φ is true in $\mathcal{A}$ under the assignment α) inductively:

- $\mathcal{A} \not\models_\alpha \perp$
- $\mathcal{A} \models_\alpha t_1 = t_2$ iff $\alpha(t_1) = \alpha(t_2)$
- $\mathcal{A} \models_\alpha R(t_1, \ldots, t_k)$ iff $(\alpha(t_1), \ldots, \alpha(t_k)) \in R^{\mathcal{A}}$
- $\mathcal{A} \models_\alpha (\varphi \to \psi)$ iff $\mathcal{A} \not\models_\alpha \varphi$ or $\mathcal{A} \models_\alpha \psi$
- $\mathcal{A} \models_\alpha (\forall x)\varphi$ iff for every $a \in u(\mathcal{A})$, $\mathcal{A} \models_{\alpha[x \mapsto a]} \varphi$
- $\mathcal{A} \models_\alpha (\exists x)\varphi$ iff there is some $a \in u(\mathcal{A})$, $\mathcal{A} \models_{\alpha[x \mapsto a]} \varphi$

Last two cases, α has no effect on the value of x. Thus, assignments matter only to free variables.

Example of Satisfaction

- $\mathcal{G} = (\{1,2,3\}, E^{\mathcal{G}} = \{(1,2),(2,1),(2,3),(3,2)\})$
  - For any assignment α, $\mathcal{G} \models_{\alpha} \forall x \forall y (E(x,y) \rightarrow E(y,x))$ iff for every $a, b \in \{1,2,3\}$, $\mathcal{G} \models_{\alpha[x \mapsto a, y \mapsto b]} (E(x,y) \rightarrow E(y,x))$

Example of Satisfaction

- $\mathcal{G} = (\{1,2,3\}, E^{\mathcal{G}} = \{(1,2),(2,1),(2,3),(3,2)\})$
  - For any assignment α, $\mathcal{G} \models_\alpha \forall x \forall y (E(x,y) \to E(y,x))$ iff for every $a, b \in \{1,2,3\}$, $\mathcal{G} \models_{\alpha[x \mapsto a, y \mapsto b]} (E(x,y) \to E(y,x))$
  - There is an assignment α which satisfies $\mathcal{G} \models_\alpha \exists x (E(x,y) \land E(x,z) \land y \neq z)$

Example of Satisfaction

- $\mathcal{G} = (\{1,2,3\}, E^{\mathcal{G}} = \{(1,2),(2,1),(2,3),(3,2)\})$
  - For any assignment α, $\mathcal{G} \models_\alpha \forall x \forall y (E(x,y) \to E(y,x))$ iff for every $a, b \in \{1,2,3\}$, $\mathcal{G} \models_{\alpha[x \mapsto a, y \mapsto b]} (E(x,y) \to E(y,x))$
  - There is an assignment α which satisfies $\mathcal{G} \models_\alpha \exists x (E(x,y) \wedge E(x,z) \wedge y \neq z)$
  - $\mathcal{G} \not\models \exists x \forall y E(x,y)$, $\mathcal{G} \models \forall x \exists y E(x,y)$
- $\mathcal{W} = abaaa$

Example of Satisfaction

- $\mathcal{G} = (\{1,2,3\}, E^{\mathcal{G}} = \{(1,2),(2,1),(2,3),(3,2)\})$
  - For any assignment α, $\mathcal{G} \models_\alpha \forall x \forall y (E(x,y) \rightarrow E(y,x))$ iff for every $a, b \in \{1,2,3\}$, $\mathcal{G} \models_{\alpha[x \mapsto a, y \mapsto b]} (E(x,y) \rightarrow E(y,x))$
  - There is an assignment α which satisfies $\mathcal{G} \models_\alpha \exists x (E(x,y) \wedge E(x,z) \wedge y \neq z)$
  - $\mathcal{G} \not\models \exists x \forall y E(x,y)$, $\mathcal{G} \models \forall x \exists y E(x,y)$
- $\mathcal{W} = abaaa$
  - There is an assignment α for which $\mathcal{W} \models_\alpha (Q_a(x) \wedge Q_a(y) \wedge S(x,y))$

Example of Satisfaction

- $\mathcal{G} = (\{1,2,3\}, E^{\mathcal{G}} = \{(1,2),(2,1),(2,3),(3,2)\})$
  - For any assignment α, $\mathcal{G} \models_\alpha \forall x \forall y (E(x,y) \rightarrow E(y,x))$ iff for every $a, b \in \{1,2,3\}$, $\mathcal{G} \models_{\alpha[x \mapsto a, y \mapsto b]} (E(x,y) \rightarrow E(y,x))$
  - There is an assignment α which satisfies $\mathcal{G} \models_\alpha \exists x (E(x,y) \wedge E(x,z) \wedge y \neq z)$
  - $\mathcal{G} \not\models \exists x \forall y E(x,y)$, $\mathcal{G} \models \forall x \exists y E(x,y)$
- $\mathcal{W} = abaaa$
  - There is an assignment α for which $\mathcal{W} \models_\alpha (Q_a(x) \wedge Q_a(y) \wedge S(x,y))$
  - There is no assignment α which satisfies $\exists x \exists y (Q_b(x) \wedge Q_b(y) \wedge x \neq y)$

Example of Satisfaction

- $\mathcal{G} = (\{1,2,3\}, E^{\mathcal{G}} = \{(1,2),(2,1),(2,3),(3,2)\})$
    - For any assignment α, $\mathcal{G} \models_\alpha \forall x \forall y (E(x,y) \rightarrow E(y,x))$ iff for every $a, b \in \{1,2,3\}$, $\mathcal{G} \models_{\alpha[x \mapsto a, y \mapsto b]} (E(x,y) \rightarrow E(y,x))$
    - There is an assignment α which satisfies $\mathcal{G} \models_\alpha \exists x (E(x,y) \wedge E(x,z) \wedge y \neq z)$
    - $\mathcal{G} \not\models \exists x \forall y E(x,y)$, $\mathcal{G} \models \forall x \exists y E(x,y)$
- $\mathcal{W} = abaaa$
    - There is an assignment α for which $\mathcal{W} \models_\alpha (Q_a(x) \wedge Q_a(y) \wedge S(x,y))$
    - There is no assignment α which satisfies $\exists x \exists y (Q_b(x) \wedge Q_b(y) \wedge x \neq y)$
    - Prove or disprove : $\mathcal{W} \models \exists x \forall y [Q_b(x) \wedge x < y \wedge Q_a(y)]$
    - Prove or disprove : $\mathcal{W} \models \exists x \forall y [Q_b(x) \wedge x < y \Rightarrow Q_a(y)]$

Satisfiability, Validity and Equivalence

- A formula φ over a signature τ is said to be satisfiable iff for some τ-structure $\mathcal{A}$ and assignment α, $\mathcal{A} \models_\alpha \varphi$

Satisfiability, Validity and Equivalence

- A formula φ over a signature τ is said to be satisfiable iff for some τ-structure $\mathcal{A}$ and assignment α, $\mathcal{A} \models_\alpha \varphi$
- A formula φ over a signature τ is said to be valid iff for every τ-structure $\mathcal{A}$ and assignment α, $\mathcal{A} \models_\alpha \varphi$

Satisfiability, Validity and Equivalence

- A formula φ over a signature τ is said to be satisfiable iff for some τ-structure $\mathcal{A}$ and assignment α, $\mathcal{A} \models_\alpha \varphi$
- A formula φ over a signature τ is said to be valid iff for every τ-structure $\mathcal{A}$ and assignment α, $\mathcal{A} \models_\alpha \varphi$
- Formulae $\varphi(x_1, \ldots, x_n)$ and $\psi(x_1, \ldots, x_n)$ are equivalent denoted $\varphi \equiv \psi$ iff for every $\mathcal{A}$ and α, $\mathcal{A} \models_\alpha \varphi$ iff $\mathcal{A} \models_\alpha \psi$

Satisfiability, Validity and Equivalence

- A formula φ over a signature τ is said to be satisfiable iff for some τ-structure $\mathcal{A}$ and assignment α, $\mathcal{A} \models_\alpha \varphi$
- A formula φ over a signature τ is said to be valid iff for every τ-structure $\mathcal{A}$ and assignment α, $\mathcal{A} \models_\alpha \varphi$
- Formulae $\varphi(x_1, \ldots, x_n)$ and $\psi(x_1, \ldots, x_n)$ are equivalent denoted $\varphi \equiv \psi$ iff for every $\mathcal{A}$ and α, $\mathcal{A} \models_\alpha \varphi$ iff $\mathcal{A} \models_\alpha \psi$
- Recap $\varphi_1(x) = \forall y R(x, y)$ and $\varphi_2 = \exists x \forall y R(x, y)$.
- It is clear that whenever φ_2 is satisfiable on $\mathcal{A}$, one can find an assignment α such that $\mathcal{A} \models_\alpha \varphi_1(x)$.
- Likewise, if $\mathcal{A} \models_\alpha \varphi_1(x)$, then $\mathcal{A} \models \varphi_2$.
- Thus, $\varphi_1(x), \varphi_2$ agree on satisfiability.

True or False?

For a formula φ and assignments α_1 and α_2 such that for every $x \in \mathit{free}(\varphi)$, $\alpha_1(x) = \alpha_2(x)$, $\mathcal{A} \models_{\alpha_1} \varphi$ iff $\mathcal{A} \models_{\alpha_2} \varphi$

True or False?

For a formula φ and assignments α_1 and α_2 such that for every $x \in \mathit{free}(\varphi)$, $\alpha_1(x) = \alpha_2(x)$, $\mathcal{A} \models_{\alpha_1} \varphi$ iff $\mathcal{A} \models_{\alpha_2} \varphi$

- For example, $\varphi(y) = \forall x(R(x, y) \to \forall z P(z))$

True or False?

For a formula φ and assignments α_1 and α_2 such that for every $x \in free(\varphi)$, $\alpha_1(x) = \alpha_2(x)$, $\mathcal{A} \models_{\alpha_1} \varphi$ iff $\mathcal{A} \models_{\alpha_2} \varphi$

- For example, $\varphi(y) = \forall x(R(x, y) \to \forall z P(z))$
- Consider two assignments α_1, α_2 such that $\alpha_1(y) = \alpha_2(y) = \alpha(say)$

True or False?

For a formula φ and assignments α_1 and α_2 such that for every $x \in \mathit{free}(\varphi)$, $\alpha_1(x) = \alpha_2(x)$, $\mathcal{A} \models_{\alpha_1} \varphi$ iff $\mathcal{A} \models_{\alpha_2} \varphi$

- For example, $\varphi(y) = \forall x(R(x, y) \to \forall z P(z))$
- Consider two assignments α_1, α_2 such that $\alpha_1(y) = \alpha_2(y) = \alpha(\mathit{say})$
- Evaluate for all $a, b \in u(\mathcal{A})$, $R(a, \alpha) \to P(b)$

True or False?

For a formula φ and assignments α_1 and α_2 such that for every $x \in free(\varphi)$, $\alpha_1(x) = \alpha_2(x)$, $\mathcal{A} \models_{\alpha_1} \varphi$ iff $\mathcal{A} \models_{\alpha_2} \varphi$

- For example, $\varphi(y) = \forall x(R(x, y) \to \forall z P(z))$
- Consider two assignments α_1, α_2 such that $\alpha_1(y) = \alpha_2(y) = \alpha(say)$
- Evaluate for all $a, b \in u(\mathcal{A})$, $R(a, \alpha) \to P(b)$
- $\mathcal{A} \models_{\alpha_1} \varphi$ iff $\mathcal{A} \models_{\alpha_2} \varphi$

True or False?

For a sentence φ, and any two assignments α_1 and α_2, $\mathcal{A} \models_{\alpha_1} \varphi$ iff $\mathcal{A} \models_{\alpha_2} \varphi$

True or False?

For a sentence φ, and any two assignments α_1 and α_2, $\mathcal{A} \models_{\alpha_1} \varphi$ iff $\mathcal{A} \models_{\alpha_2} \varphi$

No free variables!

Check SAT

- $\varphi = \exists x[(\forall y E(x, y)) \wedge \forall z[(\forall y E(z, y)) \rightarrow z = x]]$. Does φ evaluate to true under some graph structure?

Check SAT

- $\varphi = \exists x[(\forall y E(x,y)) \wedge \forall z[(\forall y E(z,y)) \rightarrow z = x]]$. Does φ evaluate to true under some graph structure?
- $\psi = \exists x[Q_a(x) \wedge \forall y[(y < x \wedge Q_b(y)) \rightarrow (z < x \wedge y < z \wedge Q_a(z))]]$. Does ψ evaluate to true under some word structure?