Informatics 1 - Computation & Logic: Tutorial 5 v:1.12 1st November 2017 16:36:40

Satisfiability and Resolution

Week 7: 30 October–3 November 2017

Please attempt the entire worksheet in advance of the tutorial, and bring all work with you. Tutorials cannot function properly unless you study the material in advance. Attendance at tutorials is **obligatory**; please let the ITO know if you cannot attend.

You may work with others, indeed you should do so; but you must develop your own understanding; you can't phone a friend during the exam. If you do not master the coursework you are unlikely to pass the exams.

We say that $\Gamma \models \Delta$ iff every state that satisfies every expression in Γ satisfies at least one expression in Δ. In particular, $\models \Delta$ (the LHS is empty) iff $\bigvee \Delta$ is a tautology, and $\Sigma \models$ (the RHS is empty) iff Σ is not satisfiable.

We represent a state as (the conjunction of) the set V of literals true in that state. For any formula ϕ either $V \models \phi$ or $V \models \neg\phi$ — as V makes ϕ either true or false.

$$\text{So, } \Gamma \models \Delta \textbf{ iff for all states } V.$$
$$\big((\textbf{for all } \varphi \in \Gamma.\ V \models \varphi) \Rightarrow (\textbf{for some } \psi \in \Delta.\ V \models \psi)\big)$$
$$\Gamma \not\models \Delta \textbf{ iff there is a counterexample } V \textbf{ such that}$$
$$(\textbf{for all } \varphi \in \Gamma.\ V \models \varphi) \textbf{ and, } (\textbf{for all } \psi \in \Delta.\ V \models \neg\psi)$$

Homework

1. Which of the following statements are true?
   In each false case give a counterexample (specify values for Γ, Δ that make one entailment valid, then give values for φ, ψ that show the other is invalid).

(a)	$\Gamma \models \Delta, \varphi$	**iff**	$\Gamma, \neg\varphi \models \Delta$	**true**
(b)	$\Gamma \models \Delta, \neg\varphi$	**iff**	$\Gamma, \varphi \models \Delta$	**true**
(c)	$\Gamma \not\models \Delta, \varphi \wedge \psi$	**iff**	$\Gamma \models \Delta, \varphi, \psi$	$\Gamma = \Delta = \emptyset; \varphi = \top; \psi = \bot$
(d)	$\Gamma \models \Delta, \varphi \vee \psi$	**iff**	$\Gamma \models \Delta, \varphi, \psi$	**true**
(e)	$\Gamma, \varphi \vee \psi \not\models \Delta$	**iff**	$\Gamma, \varphi, \psi \models \Delta$	$\Gamma = \emptyset; \Delta = \{\psi\}; \varphi = \top; \psi = \bot$
(f)	$\Gamma, \varphi \wedge \psi \models \Delta$	**iff**	$\Gamma, \varphi, \psi \models \Delta$	**true**

You should have found that 1(a) is true: $\Gamma \models \Delta, \varphi$ **iff** $\Gamma, \neg\varphi \models \Delta$.
For example (in this example $\Delta = \emptyset$),

$$D \to (A \vee B), E \vee (C\ ?\ \neg A : B), (B\ ?\ E \to D : C \vee E) \models (D \wedge C) \to (B \vee E)$$

$$\textbf{iff}$$

$$D \to (A \vee B), E \vee (C\ ?\ \neg A : B), (B\ ?\ E \to D : C \vee E), \neg\big((D \wedge C) \to (B \vee E)\big) \models$$

In general, if we write $\Delta^{\neg}$ for $\{\neg\psi \mid \psi \in \Delta\}$, $\Gamma \models \Delta$ **iff** $\Gamma, \Delta^{\neg} \models$
This means that, $\Gamma \models \Delta$ **iff** $\Gamma, \Delta^{\neg}$ is inconsistent.

Resolution is a procedure used to search for a state that satisfies a given set of constraints, Σ. If there is no state satisfying Σ — which means that Σ is inconsistent — then resolution will produce the empty clause.

> So, we can test whether $\Gamma \models \Delta$ by converting $\Sigma = \Gamma, \Delta^{\neg}$ to CNF and then using resolution. If resolution of Σ produces the empty clause then the entailment is valid; otherwise a satisfying valuation for Σ provides a counterexample to the entailment.

In the examples below there is only one conclusion — Δ is a singleton.

2. Use resolution to determine whether the following entailment is valid.

$$A \to (B \vee C), \neg D \to \neg(B \wedge C) \models A \to D$$

(a) First convert each assumption and the negation of each conclusion to clausal form.

i. $A \to (B \vee C)$ $\{\neg A, B, C\}$
ii. $\neg D \to \neg(B \wedge C)$ $\{\neg B, \neg C, D\}$
iii. $\neg(A \to D)$ $\{A\}$,$\{\neg D\}$

(b) Check whether this set of constraints is consistent:

When our constraints include singleton clauses (also known as *unit clauses*), then the literals they include **must** be true in any satisfying assignment.
We can simlify our clauses by making these literals true.
For example, here,$\{A\}$ and $\{\neg D\}$ are unit clauses.

	AD	B		
$\{\neg A, B, C\}$ $\{\neg B, \neg C, D\}\}$ $\{A\}$ $\{\neg D\}$	$\{B, C\}$ $\{\neg B, \neg C\}$	$\cancel{\{C, \neg C\}}$		

The constraints are consistent so the entailment is invalid.
To make the unit clauses true, we made A true and D false; this makes the conclusion of the entailment false To construct a counterexample, we can choose any value for C, and make $B = \neg C$; it is easy to check that this makes the assumptions true

3. Determine whether the following entailment is valid

$$(D \to (A \vee B)), (E \vee (C \ ? \ \neg A : D)), (B \ ? \ C \to E : A \to C) \models (B \to C) \to (D \to E)$$

(a) First convert each assumption and the negation of each conclusion to clausal form.

i. $D \to (A \vee B)$ $\{A, B, \neg D\}$

ii. $E \vee (C \ ? \ \neg A : D)$ $\{\neg A, \neg C, E\}, \{D, C, E\}$

iii. $(B \ ? \ C \to E : A \to C)$ $\{\neg B, \neg C, E\}$,$\{\neg A, B, C\}$

iv. $\neg\big((B \to C) \to (D \to E)\big)$ $\{\neg B, C\}, \{D\}, \{\neg E\}$

(b) Check whether this set of constraints is consistent:
First, make unit literals true, $\top$.

DE	A	B	C	
${}^{D}\{A, B, \neg D\}$	${}^{A}\{A, B\}$			
${}^{E}\{\neg A, \neg C, E\}$	${}^{A}\{\neg A, \neg C\}$	${}^{B}\{B, \neg C\}$	${}^{C}\{\neg C\}$	
${}^{D}\{D, C, E\}$				
${}^{E}\{\neg B, \neg C, E\}$	${}^{B}\{\neg B, \neg C\}$			
${}^{A}\{\neg A, B, C\}$		${}^{B}\{B, C\}$	${}^{C}\{C\}$	$\{\}$
${}^{B}\{\neg B, C\}$				
${}^{D}\{D\}$				
${}^{E}\{\neg E\}$				

The constraints are inconsistent so the entailment is valid.

You should now be able to tackle many of the questions on past papers.

These can be found at `https://www.inf.ed.ac.uk/teaching/exam_papers/`

Here are some questions you should be able to answer:

Dec 2014 Q1, Q3

Dec 2015 Q1, Q3(a,b)

Aug 2016 Q1, Q3(a,b)

Dec 2016 Q1, Q2, Q4

4. Try these and bring any questions you may have to your tutorial, or visit InfBase.

Entailment

$$\Gamma \models \Delta \quad \textbf{iff for all states } V.$$
$$\big((\textbf{for all } \varphi \in \Gamma.\ V \models \varphi) \Rightarrow (\textbf{for some } \psi \in \Delta.\ V \models \psi)\big)$$

$$\Gamma \not\models \Delta \quad \textbf{iff there is a counterexample } V \textbf{ such that}$$
$$(\textbf{for all } \varphi \in \Gamma.\ V \models \varphi) \textbf{ and, } (\textbf{for all } \psi \in \Delta.\ V \models \neg\psi)$$

We will normally omit the set-brackets $\{\}$, and other set-notation, when writing entailments. For example, if φ, ψ, θ are expressions, we write

$\Gamma \models \varphi$	**for**	$\Gamma \models \{\varphi\}$
$\Gamma, \theta \models \varphi, \psi$	**for**	$\Gamma \cup \{\theta\} \models \{\varphi, \psi\}$

Now consider what this definition of $\models$ means in some special cases.

- The case where Δ is empty.
  $\Gamma \models$
  $(\Gamma \models \emptyset)$ means that the conjunction $\bigwedge \Gamma$ of all expressions in Γ is contradictory.
- The case where Γ is empty.
  $\models \Delta$
  $(\emptyset \models \Delta)$ means that the disjunction $\bigvee \Delta$ of all expressions in Δ is a tautology.
- The case in which Γ is a non-contradictory set of literals.
  A non-contradictory set of literals determines a truth valuation of the atoms it mentions.
  In this case, $\Gamma \models \varphi$ means that the valuation determined by $\bigwedge \Gamma$ makes φ true.
- If both Γ and Δ are sets of literals, and Γ is non-contradictory,
  we can view Γ as representing a valuation V and Δ as representing a clause
  $\Gamma \models \Delta$ **iff** $\Gamma \cap \Delta$ is non-empty: $(\Gamma \cap \Delta \neq \emptyset)$, which means that V satisfies Δ.

Tutorial Activity

1. As usual, you should start by working as a group to identify and resolve any problems with the homework.

2. Consider the simple Traffic Light, introduced in class, with a cycle of four states. The four legal states of the light are captured by the four states of the variables R, A representing the red and amber lights. The Green light is on only in the state represented by $\neg R \wedge \neg A$.

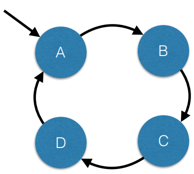
   In this question, you will extend the model by adding one more state variable C, representing the presence of a car – there is normally a sensor in the road that indicates a car is passing or waiting at the light, C could be the output of this sensor. The new system will have eight states – each each legal state of the lights, with a car present C or absent $\neg C$.

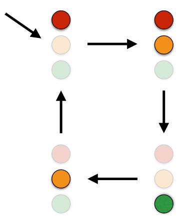
   (a) Assume that the lights cycle through the proper sequence, and that cars always obey the traffic regulations, draw a diagram of the legal state transitions. Give a logical description of the legal transitions.

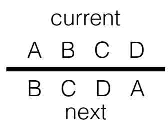
   (b) Can you remove some transitions to describe a system in which the lights will not change to green unless there is a car waiting? Give a logical description of the transitions you allow.

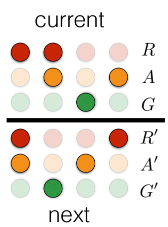
   (c) How might you model an intersection with two pairs of lights? What should be the states of this system? What conditions on the states and transitions do we need to specify a safe and efficient system?

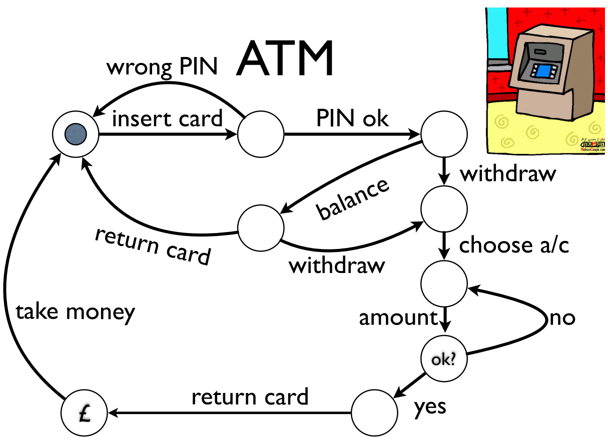
3. Modify the PIN-checking section of the ATM to allow more than one attempt to enter the PIN. Your machine should retain the card after three wrong PIN entries.











This tutorial exercise sheet was written by Dave Cochran and Michael Fourman, drawing on material from an earlier tutorials produced by Paolo Besana, Thomas French, and Areti Manataki. Send comments to `Michael.Fourman@ed.ac.uk`