Undecidability of first-order logic

Ana Sokolova UNIVERSITY of SALZBURG

University of Vienna 9.10.18

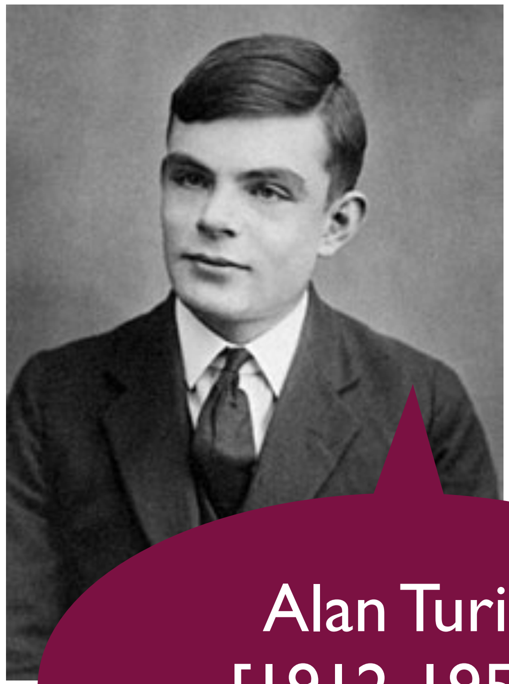
www.cs.uni-salzburg.at/~anas/undec-FOL

Decidability

Alonzo Church
[1903-1995]
1935-36

study motivated by Hilbert's problems, in particular by the question of decidability of FOL

A problem (set) P is decidable if there exists an algorithm that returns YES on any p ∈ P and NO otherwise.



Alan Turing
[1912-1954]
1936-37

decision procedure
= YES/NO oracle
= computable function
= Turing machine

Hilbert's Entscheidungsproblem

1928

Is there an algorithm that decides whether arbitrary FOL formula is valid / satisfiable ?

Is VALID = {φ | φ is a valid first-order formula} decidable ?

Clearly, VALID is decidable iff
SAT = {φ | φ is a satisfiable first-order formula} is.

The answer is NO

Outline of Turing's proof

Julius Richard Büchi
[1924 -1984]
1961

Turing introduced the Turing machines and proved first:

Reduction from the halting problem to unsatisfiability

Lecture notes by Larry Moss and Guram Bezhanishvili 2008

Theorem HALT

The set

HALT = { (M, i) | M is a Turing machine that halts on input i}

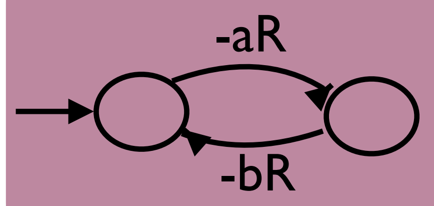
is undecidable.

Theorem HALT-e

The set

HALT-e = { M | M is a Turing machine that halts on empty input}

is undecidable.



$\longmapsto$ $\exists x \forall y \ldots \wedge (\forall x \forall z .. \vee \forall x \exists y ..) \wedge \ldots$

FOL: Reduction

HALT-e
to
UNSAT

Reduction Theorem

For a Turing machine M, we construct a FOL formula φ_M such that
φ_M is unsatisfiable iff M halts on empty input.

Corollary - Undecidability of FOL

UNSAT = $\{\varphi \mid \varphi$ is an unsatisfiable first-order formula $\}$ is undecidable.
Hence, VALID is undecidable too.

What would happen to
HALT-e if UNSAT were
decidable ?

Turing machine

= finite control (automaton states)
+ (potentially) infinite tape
+ head for reading/writing

unlimited access
to infinite memory

Turing machines

A Turing machine is a tuple $M = (Q, \Sigma, \delta, q_0, q_1)$, where

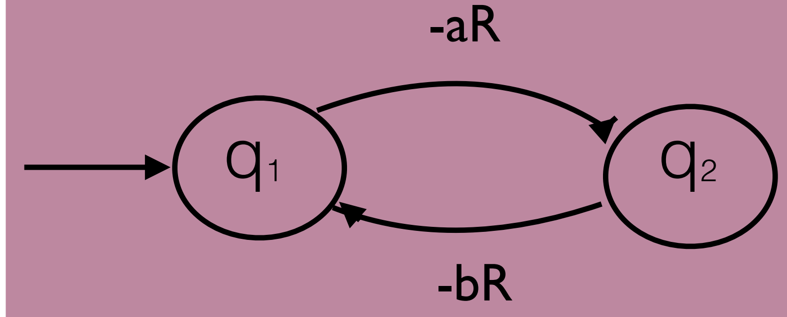
Q is a finite set of states with q_0 the halting state and q_1 the starting state, ($q_0, q_1 \in Q$)

Σ is a finite set, the alphabet, and

$\delta: Q \setminus \{q_0\} \times \Sigma \longrightarrow \Sigma \times \{L, C, R\} \times Q$ is the transition function.

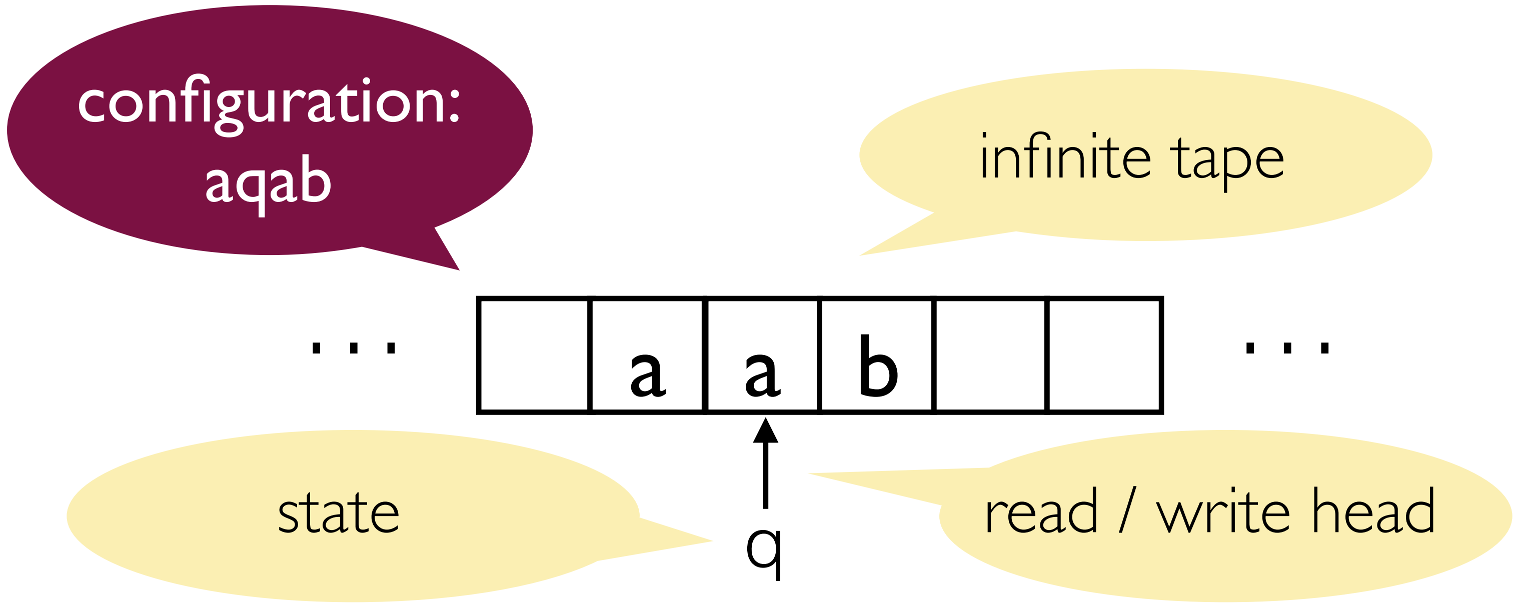
$\delta(q,a) = (b,X,r)$ means that in a state q, reading the symbol a from the cell on which the head is positioned, the TM writes b in place of a, moves the head for one cell in direction X, and changes to state r.

Example



Turing machines

Compute via configurations, triples (q,w,h)
notation uqv where uv = w and |u| = h-1.



$uaqbv \vDash uracv$ iff $\delta(q,b) = (c,L,r)$
$uaqbv \vDash uarcv$ iff $\delta(q,b) = (c,C,r)$
$uqbv \vDash ucrv$ iff $\delta(q,b) = (c,R,r)$
$qbv \vDash r\square cv$ iff $\delta(q,b) = (c,L,r)$
$qbv \vDash rcv$ iff $\delta(q,b) = (c,C,r)$
$uaq \vDash uarc$ iff $\delta(q,\square) = (c,C,r)$
$uaq \vDash uacr$ iff $\delta(q,\square) = (c,R,r)$

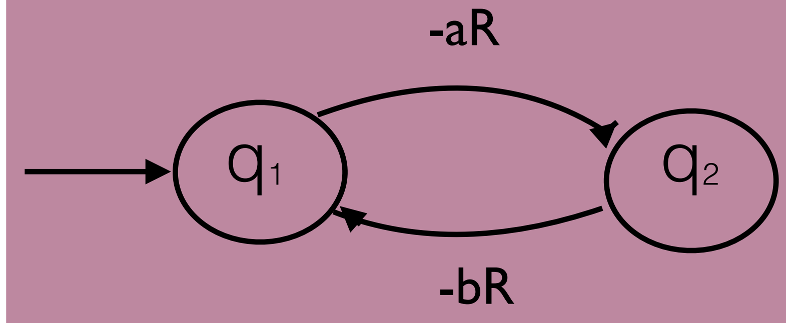
M decides P iff

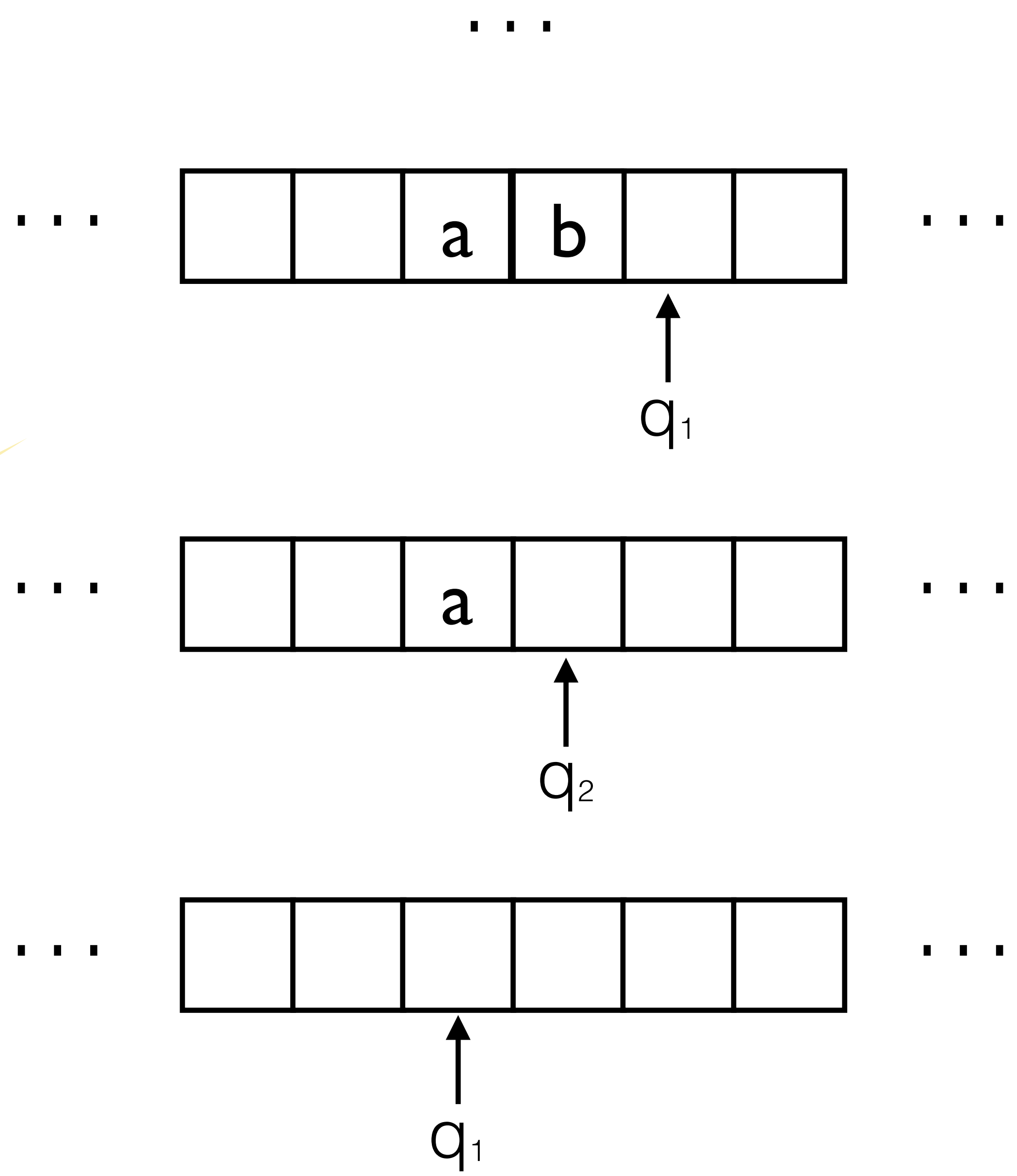
for all $p \in P$, $q_1 p \vDash^* q_0 YES$
for all $p \notin P$, $q_1 p \vDash^* q_0 NO$

sequence of configurations

Example run on empty input

Example

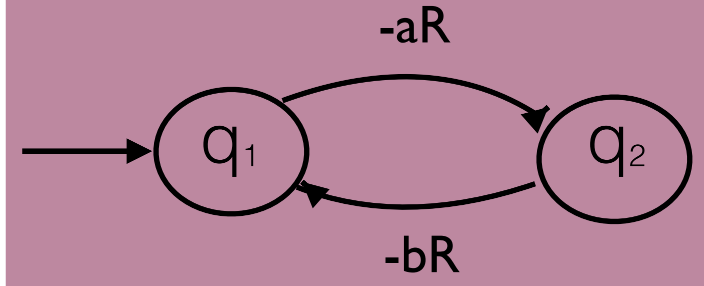




gives us an intended model

The intended model of φ_M

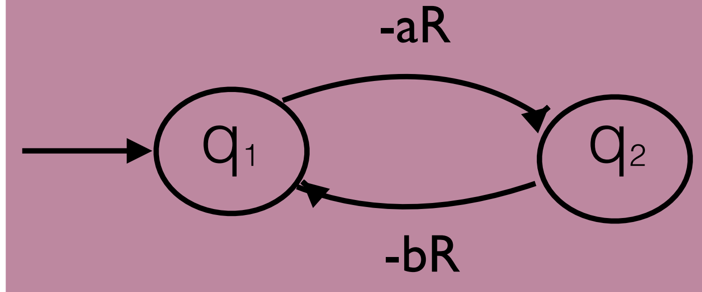
Example

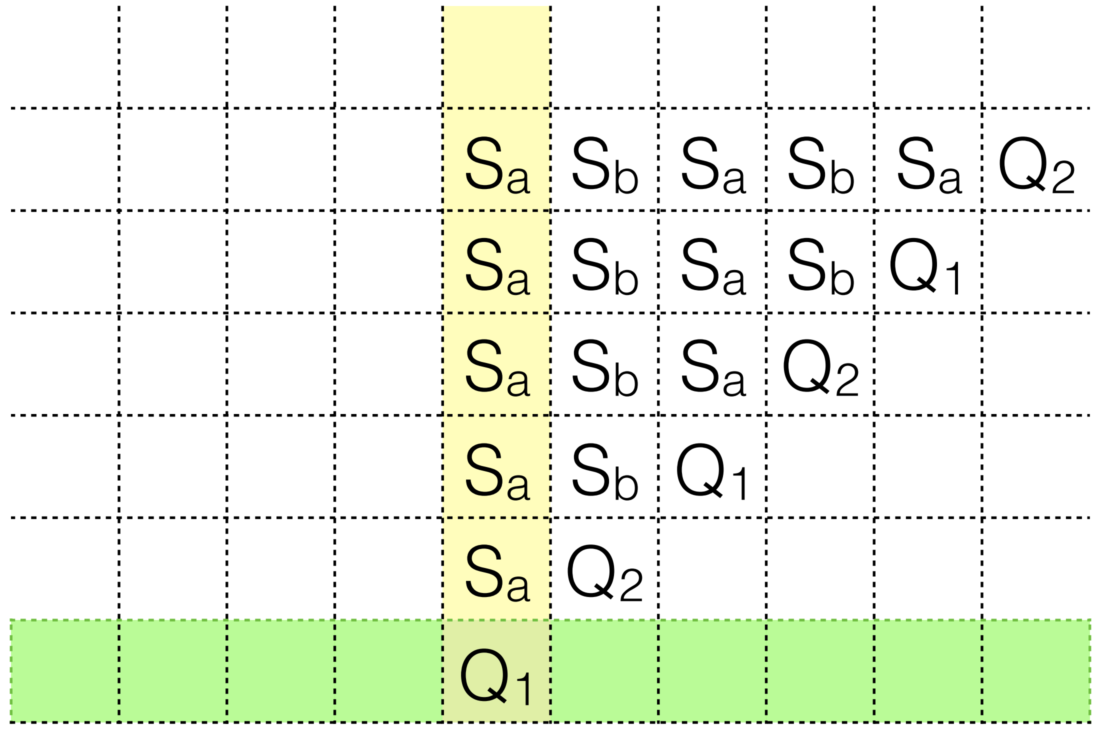


					. . .						
					S_a	S_b	S_a	S_b	S_a	Q_2	
					S_a	S_b	S_a	S_b	Q_1		
					S_a	S_b	S_a	Q_2			
					S_a	S_b	Q_1				
					S_a	Q_2					
. . .					Q_1						. . .

The signature of φ_M

Example





Q_i - unary predicate symbol for each $q_i \in Q$

Sa - unary predicate symbol for each $a \in \Sigma$

head - unary predicate symbol

+ some more function and predicate symbols for describing the upper half plane:

    origin - constant symbol

    east, west, north - unary function symbols

    y-axis, left-side, right-side, bottom - unary predicate symbols

The encoding φ_M

φ_M = upper-half-plane ∧
unique-symbol-per-cell ∧
initial-situation ∧
no-head-no-change ∧
does-not-halt ∧

one-step-formulas ∧

head-movement-formula

The encoding φM

φ_M =

upper-half-plane ∧
unique-symbol-per-cell ∧
initial-situation ∧
no-head-no-change ∧
does-not-halt ∧

one-step-formulas ∧

head-movement-formula

$$\forall x. \bigvee_{a \in \Sigma} (S_a(x) \wedge \bigwedge_{b \neq a} \neg S_b(x))$$

The encoding φ_M

φ_M =

upper-half-plane ∧
unique-symbol-per-cell ∧
initial-situation ∧
no-head-no-change ∧
does-not-halt ∧

$$Q_1(\text{origin}) \wedge \bigwedge_{k \neq 1} \neg Q_k(\text{origin}) \wedge$$
$$\forall x.\text{bottom}(x) \rightarrow S_{\Box}(x) \wedge (x = \text{origin} \leftrightarrow \text{head}(x))$$

one-step-formulas ∧

head-movement-formula

The encoding φ_M

φ_M = upper-half-plane ∧
unique-symbol-per-cell ∧
initial-situation ∧
no-head-no-change ∧
does-not-halt ∧

one-step-formulas ∧

head-movement-formula

Homework

The encoding φ_M

φ_M = upper-half-plane $\wedge$
unique-symbol-per-cell $\wedge$
initial-situation $\wedge$
no-head-no-change $\wedge$
does-not-halt $\wedge$

one-step-formulas $\wedge$

head-movement-formula

$\forall x. \neg Q_0(x)$

The encoding φ$_M$

φ$_M$ = upper-half-plane ∧
unique-symbol-per-cell ∧
initial-situation ∧
no-head-no-change ∧
does-not-halt ∧

one-step-formulas ∧

head-movement-formula

$\delta(q_i,a) = (b,L,q_j)$

$$\forall x.(\text{head}(x) \land Q_i(x) \land S_a(x)) \rightarrow$$
$$(S_b(\text{north}(x)) \land$$
$$\text{head}(\text{north} \circ \text{west}(x)) \land$$
$$Q_j(\text{north} \circ \text{west}(x))$$

$$\forall x.(\text{head}(x) \land Q_i(x) \land S_a(x)) \rightarrow$$
$$\ldots$$
$$\forall x.(\text{head}(x) \land Q_i(x) \land S_a(x)) \rightarrow$$
$$\ldots$$

The encoding φM

φM = upper-half-plane ∧
unique-symbol-per-cell ∧
initial-situation ∧
no-head-no-change ∧
does-not-halt ∧

one-step-formulas ∧

head-movement-formula

$$\begin{aligned}
&\forall x.\mathrm{head}(x) \wedge Q_i(x) \wedge S_a(x) \rightarrow \\
&x = \mathrm{origin} \vee \\
&\exists y.(\mathrm{north} \circ \mathrm{west}(y) = x \wedge \\
\bigvee_{\delta(q_j,b)=(c,L,q_i)} &(\mathrm{head}(y) \wedge Q_j(y) \wedge S_b(y))) \\
&\vee \ (\mathrm{north}(y) = x \wedge \cdots) \\
&\vee \ (\mathrm{north} \circ \mathrm{east}(y) = x \wedge \cdots))
\end{aligned}$$

The encoding φ_M

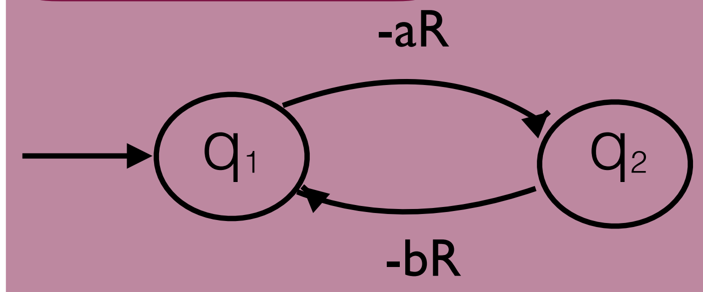
φ_M = upper-half-plane ∧
unique-symbol-per-cell ∧
initial-situation ∧
no-head-no-change ∧
does-not-halt ∧

one-step-formulas ∧

head-movement-formula

Let's write φ_M for

Example



The encoding φ_M

Next week we will prove

φ_M is satisfiable iff M does not halt on empty input.

$\Leftarrow$ easy:

the intended model is indeed a model

$\Rightarrow$ Simulation Lemma:

given a model of φ_M we can show that M does not halt on empty

www.cs.uni-salzburg.at/~anas/undec-FOL