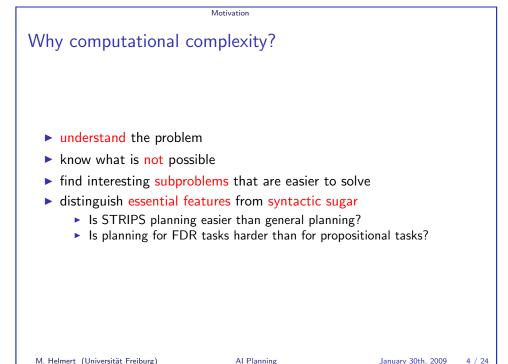
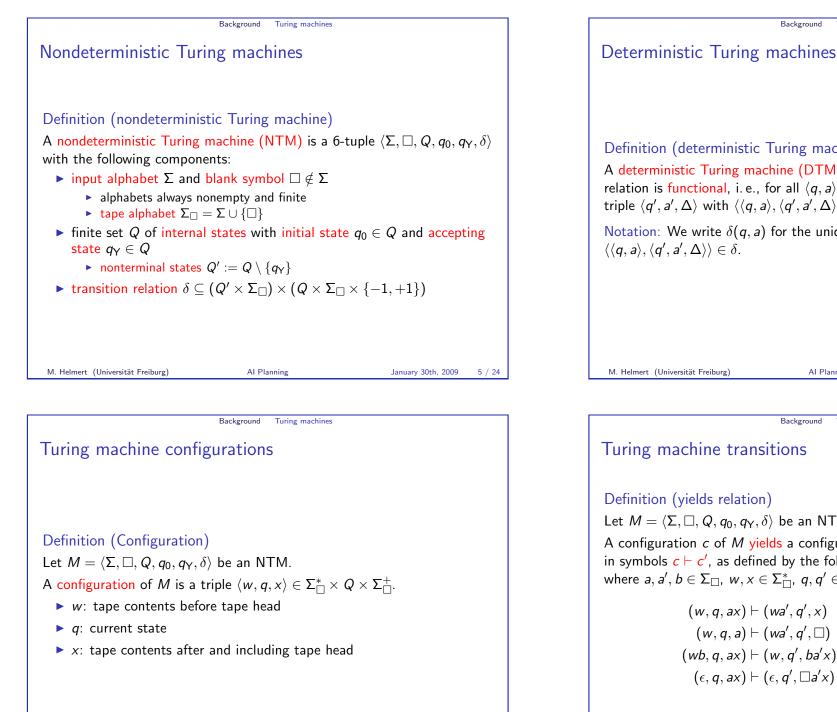


January 30th, 2009 — 13. Computational complexity of classical planning			
Motivation			
Background Turing machines Complexity classes			
Complexity of propositional planning Plan existence and bounded plan existence PSPACE-completeness			
More complexity results			
M. Helmert (Universität Freiburg)	Al Planning	January 30th, 2009	2 / 24



January 30th, 2009



Definition (deterministic Turing machine) A deterministic Turing machine (DTM) is an NTM where the transition relation is functional, i.e., for all  $\langle q, a \rangle \in Q' \times \Sigma_{\Box}$ , there is exactly one triple  $\langle q', a', \Delta \rangle$  with  $\langle \langle q, a \rangle, \langle q', a', \Delta \rangle \rangle \in \delta$ . Notation: We write  $\delta(q, a)$  for the unique triple  $\langle q', a', \Delta \rangle$  such that AI Planning January 30th, 2009 6 / 24 Background Turing machines

Turing machines

Background

# Turing machine transitions Definition (yields relation) Let $M = \langle \Sigma, \Box, Q, q_0, q_Y, \delta \rangle$ be an NTM. A configuration c of M yields a configuration c' of M, in symbols $c \vdash c'$ , as defined by the following rules, where $a, a', b \in \Sigma_{\Box}, w, x \in \Sigma_{\Box}^*, q, q' \in Q$ and $\langle \langle q, a \rangle, \langle q', a', \Delta \rangle \rangle \in \delta$ : $(w, q, ax) \vdash (wa', q', x)$ if $\Delta = +1, |x| > 1$ $(w, q, a) \vdash (wa', q', \Box)$ if $\Delta = +1$ $(wb, q, ax) \vdash (w, q', ba'x)$ if $\Delta = -1$ $(\epsilon, q, ax) \vdash (\epsilon, q', \Box a'x)$ if $\Delta = -1$

#### Background Turing machines

### Accepting configurations

#### Definition (accepting configuration, time)

Let  $M = \langle \Sigma, \Box, Q, q_0, q_Y, \delta \rangle$  be an NTM, let  $c = \langle w, q, x \rangle$  be a configuration of M, and let  $n \in \mathbb{N}_0$ .

- ▶ If  $q = q_Y$ , *M* accepts *c* in time *n*.
- If q ≠ q<sub>Y</sub> and M accepts some c' with c ⊢ c' in time n, then M accepts c in time n + 1.

Definition (accepting configuration, space)

Let  $M = \langle \Sigma, \Box, Q, q_0, q_Y, \delta \rangle$  be an NTM, let  $c = \langle w, q, x \rangle$  be a configuration of M, and let  $n \in \mathbb{N}_0$ .

- If  $q = q_Y$  and  $|w| + |x| \le n$ , *M* accepts *c* in space *n*.
- ▶ If  $q \neq q_Y$  and M accepts some c' with  $c \vdash c'$  in space n, then M accepts c in space n.

AI Planning

```
M. Helmert (Universität Freiburg)
```

January 30th, 2009

9 / 24

Backgroun Complexity classes
Time and space complexity classes
Definition (DTIME, NTIME, DSPACE, NSPACE)
Let f : N<sub>0</sub> → N<sub>0</sub>.
Complexity class DTIME(f) contains all languages accepted in time f by some DTM.
Complexity class NTIME(f) contains all languages accepted in time f by some NTM.
Complexity class DSPACE(f) contains all languages accepted in space f by some DTM.
Complexity class NSPACE(f) contains all languages accepted in space f by some DTM.

### Accepting words and languages

Definition (accepting words)

Let  $M = \langle \Sigma, \Box, Q, q_0, q_Y, \delta \rangle$  be an NTM.

*M* accepts the word  $w \in \Sigma^*$  in time (space)  $n \in \mathbb{N}_0$ iff *M* accepts  $(\epsilon, q_0, w)$  in time (space) *n*.

Special case: M accepts 
 *ϵ* in time (space) n ∈ N<sub>0</sub>
 iff M accepts (ϵ, q<sub>0</sub>, □) in time (space) n.

Definition (accepting languages)

Let  $M = \langle \Sigma, \Box, Q, q_0, q_Y, \delta \rangle$  be an NTM, and let  $f : \mathbb{N}_0 \to \mathbb{N}_0$ . M accepts the language  $L \subseteq \Sigma^*$  in time (space) fiff M accepts each word  $w \in L$  in time (space) f(|w|), and M does not accept any word  $w \notin L$  (in any time/space).

M. Helmert (Universität Freiburg)

AI Planning

January 30th, 2009 10 / 24

Background Complexity classes

Polynomial time and space classes

Let  $\mathcal{P}$  be the set of polynomials  $p : \mathbb{N}_0 \to \mathbb{N}_0$  whose coefficients are natural numbers.

Definition (P, NP, PSPACE, NPSPACE)

 $P = \bigcup_{p \in \mathcal{P}} \mathsf{DTIME}(p)$  $NP = \bigcup_{p \in \mathcal{P}} \mathsf{NTIME}(p)$  $PSPACE = \bigcup_{p \in \mathcal{P}} \mathsf{DSPACE}(p)$  $NPSPACE = \bigcup_{p \in \mathcal{P}} \mathsf{NSPACE}(p)$ 

Background Complexity classes

#### Polynomial complexity class relationships

Theorem (complexity class hierarchy)  $P \subseteq NP \subseteq PSPACE = NPSPACE$ 

#### Proof.

 $\mathsf{P}\subseteq\mathsf{NP}\text{ and }\mathsf{PSPACE}\subseteq\mathsf{NPSPACE}\text{ is obvious because deterministic}$  Turing machines are a special case of nondeterministic ones.

 $\mathsf{NP}\subseteq\mathsf{NPSPACE}$  holds because a Turing machine can only visit polynomially many tape cells within polynomial time.

PSPACE = NPSPACE is a special case of a classical result known as Savitch's theorem (Savitch 1970).

M. Helmert (Universität Freiburg)

January 30th, 2009

Complexity of planning (Bounded) plan existence

AI Planning

Plan existence vs. bounded plan existence

Theorem (reduction from PLANEX to PLANLEN) PLANEX  $\leq_{\rho}$  PLANLEN

#### Proof.

A propositional planning task with *n* state variables has a plan iff it has a plan of length at most  $2^n - 1$ .

 $\rightsquigarrow$  map instance  $\Pi$  of PLANEX to instance  $\langle \Pi, 2^n - 1 \rangle$  of PLANLEN, where *n* is the number of *n* state variables of  $\Pi$ 

 $\rightsquigarrow$  polynomial reduction

Complexity of planning (Bounded) plan existence

## The propositional planning problem

#### Definition (plan existence)

The plan existence problem (PLANEx) is the following decision problem: GIVEN: Planning task  $\Pi$ QUESTION: Is there a plan for  $\Pi$ ?

 $\rightsquigarrow$  decision problem analogue of satisficing planning

#### Definition (bounded plan existence)

 $\begin{array}{ll} \mbox{The bounded plan existence problem (PLANLEN)} \\ \mbox{is the following decision problem:} \\ \mbox{GIVEN:} & \mbox{Planning task } \Pi, \mbox{ length bound } K \in \mathbb{N}_0 \\ \mbox{QUESTION:} & \mbox{Is there a plan for } \Pi \mbox{ of length at most } K? \end{array}$ 

 $\rightsquigarrow$  decision problem analogue of optimal planning

```
M. Helmert (Universität Freiburg)
```

AI Planning

January 30th, 2009 14 / 24

Complexity of planning PSPACE-completeness

AI Planning

### Membership in PSPACE

Theorem (PSPACE membership for PLANLEN)

 $\mathrm{PLANLEN} \in \mathsf{PSPACE}$ 

#### Proof.

Show PLANLEN  $\in$  NPSPACE and use Savitch's theorem. Nondeterministic algorithm: def plan( $\langle A, I, O, G \rangle$ , K): s := I

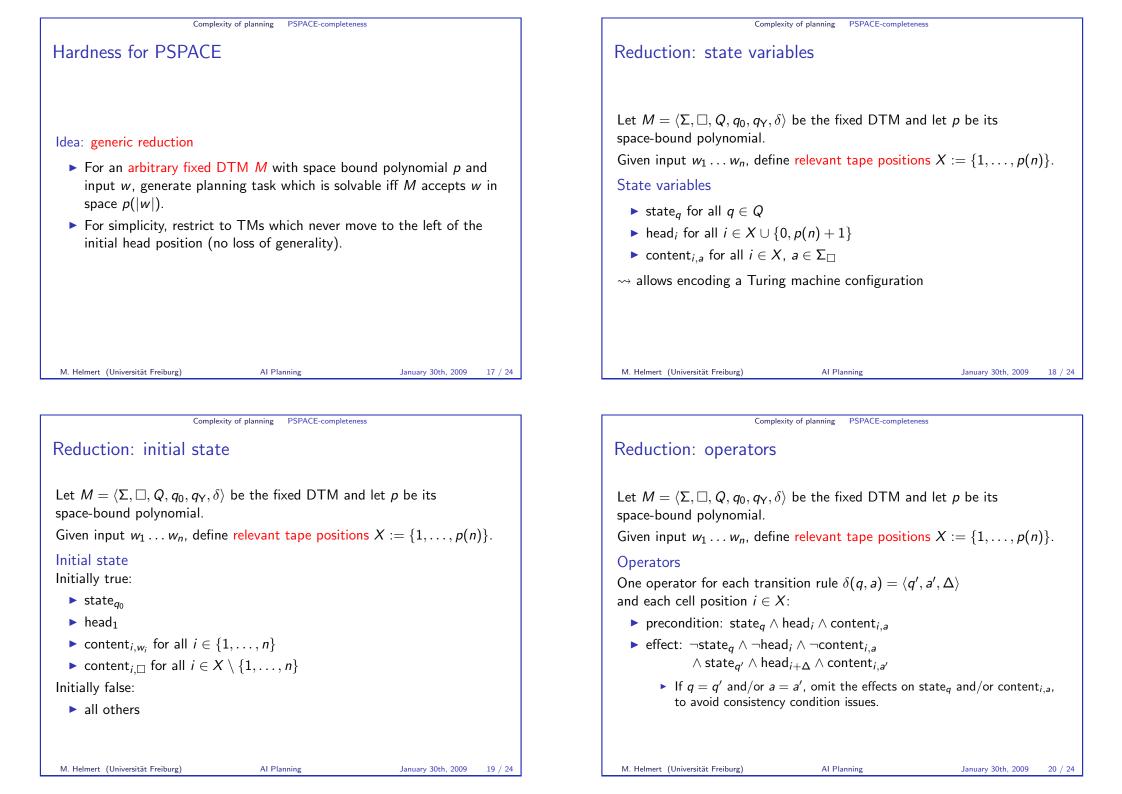
```
k := K
while s \not\models G:
guess o \in O
fail if o not applicable in s or k = 0
s := app_o(s)
k := k - 1
accept
```

accept

 $\square$ 

M. Helmert (Universität Freiburg)

13 / 24



#### Complexity of planning PSPACE-completeness

Reduction: goal

Let  $M = \langle \Sigma, \Box, Q, q_0, q_Y, \delta \rangle$  be the fixed DTM and let p be its space-bound polynomial.

Given input  $w_1 \dots w_n$ , define relevant tape positions  $X := \{1, \dots, p(n)\}$ .

Goal

 $state_{q_Y}$ 

M. Helmert (Universität Freiburg)

AI Planning

January 30th, 2009

21 / 24

23 / 24

More complexity results

#### More complexity results

In addition to the basic complexity result presented in this chapter, there are many special cases, generalizations, variations and related problems studied in the literature:

- different planning formalisms
  - e.g., finite-domain representation, nondeterministic effects, partial observability, schematic operators, numerical state variables
- syntactic restrictions of planning tasks
  - e.g., without preconditions, without conjunctive effects, STRIPS without delete effects

AI Planning

- semantic restrictions of planning task
  - e.g., restricting to certain classes of causal graphs
- particular planning domains
  - e.g., Blocksworld, Logistics, FreeCell

PSPACE-completeness for STRIPS plan existence Theorem (PSPACE-completeness; Bylander, 1994) PLANEX and PLANLEN are PSPACE-complete. This is true even when restricting to STRIPS tasks. Proof. Membership for PLANLEN was already shown. Hardness for PLANEx follows because we just presented a polynomial reduction from an arbitrary problem in PSPACE to PLANEX. (Note that the reduction only generates STRIPS tasks.) Membership for PLANEX and hardness for PLANLEN follows from the polynomial reduction from PLANEX to PLANLEN. M. Helmert (Universität Freiburg) AI Planning January 30th, 2009 22 / 24

Complexity of planning PSPACE-completeness

#### More complexity results

### Complexity results for different planning formalisms

Some results for different planning formalisms:

- ► FDR tasks:
  - same complexity as for propositional tasks ("folklore")
  - ▶ also true for the SAS<sup>+</sup> special case
- nondeterministic effects:
  - fully observable: EXP-complete (Littman, 1997)
  - unobservable: EXPSPACE-complete (Haslum & Jonsson, 1999)
  - partially observable: 2EXP-complete (Rintanen, 2004)
- schematic operators:
  - ▶ usually adds one exponential level to PLANEX complexity
  - e.g., classical case EXPSPACE-complete (Erol et al., 1995)
- numerical state variables:
  - undecidable in most variations (Helmert, 2002)