Università degli studi di Udine
Laurea Magistrale: Informatica
Lectures for April/May 2014
La verifica del software: temporal logic
Lecture 04 LTL tableau correctness

Guest lecturer: Mark Reynolds,
The University of Western Australia

May 8, 2014

Lecture 04

- Tableau for checking satisfiability in LTL continued.
- The soundness proof.
- The completeness proof
- complexity

Proof of Soundness:

(Overview first)
Use a successful tableau to make a model of the formula, thus showing that it is satisfiable.

Use a successful branch. Each STEP tells us that we are moving from one state to the next.

Within a particular state we can make all the formulas listed true there (as evaluated along the rest of the fullpath). Atomic propositions listed tell us that they are true at that state.

An induction deals with most of the rest of the formulas.

Eventualities either get satisfied and disappear in a terminating branch or have to be satisfied if the branch is ticked by the LOOP rule.

Soundness

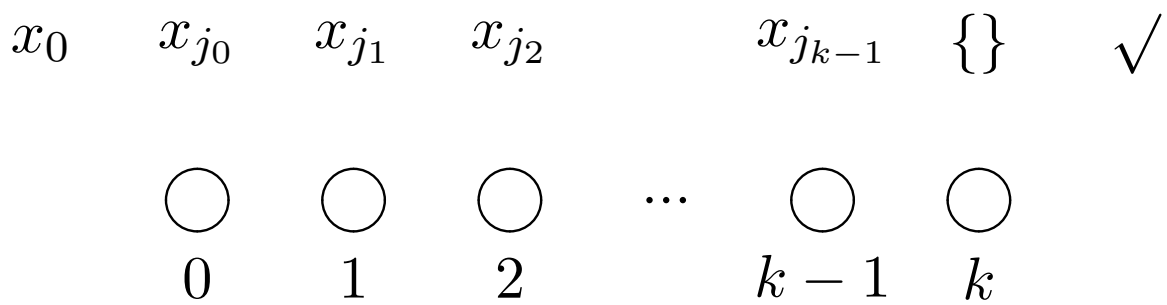
Suppose that T is a successful tableau for ϕ.
Say that the branch $b = \langle x_0, x_1, x_2, ..., x_n \rangle$ of nodes of T ends in a tick.
We build (S, R, g) from b and its tableau labels.
In fact, there are only a few x_i that matter: each time when we are about to use STEP and when we are about to use EMP or LOOP to finish (at x_n).
Let $j_0, j_1, j_2, ..., j_{k-1}$ be the indices of nodes from b at which the STEP rule is used. That is, the STEP rule is used to get from x_{j_i} to x_{j_i+1}.

Soundness continued:

$$
\begin{array}{c}
\Gamma(x_0) = \{\phi\} \\
\cdots \\
\Gamma(x_{j_0}) \\
= \\
\Gamma(x_{j_0+1}) \\
\cdots \\
\Gamma(x_{j_1}) \\
= \\
\Gamma(x_{j_1+1}) \\
\\
\vdots \\
\\
\Gamma(x_{j_{k-1}}) \\
= \\
\Gamma(x_{j_{k-1}+1}) \\
\cdots \\
\Gamma(x_n) \\
\\
\surd
\end{array}
$$

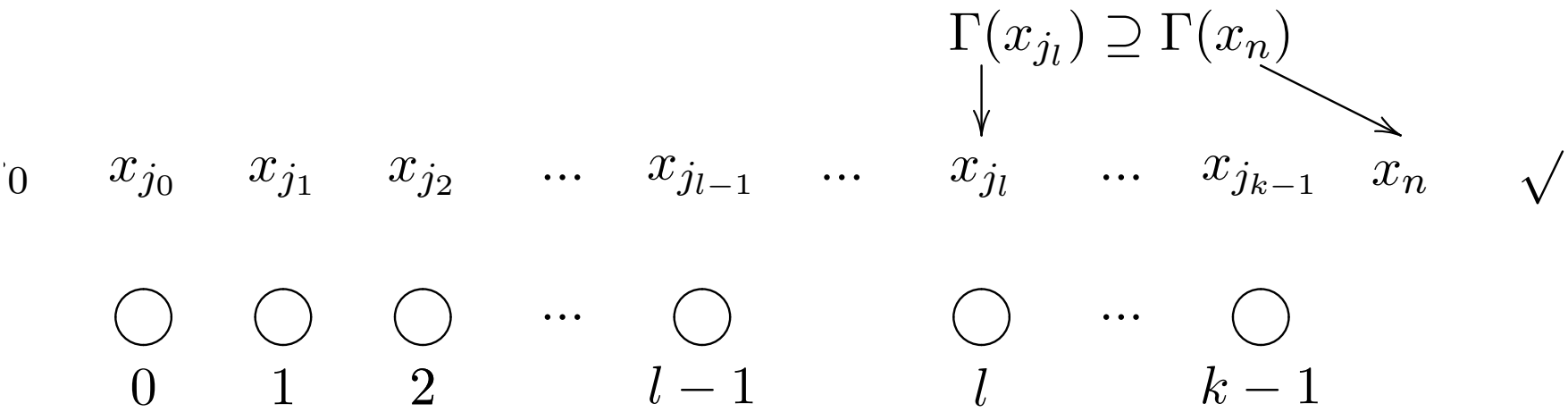
Soundness continued:

If b ends in a tick from EMP then let $S = \{0, 1, 2, ..., k\}$: so it contains $k + 1$ states. These will correspond to $x_{j_0}, x_{j_1}, ..., x_{j_{k-1}}, x_n$.



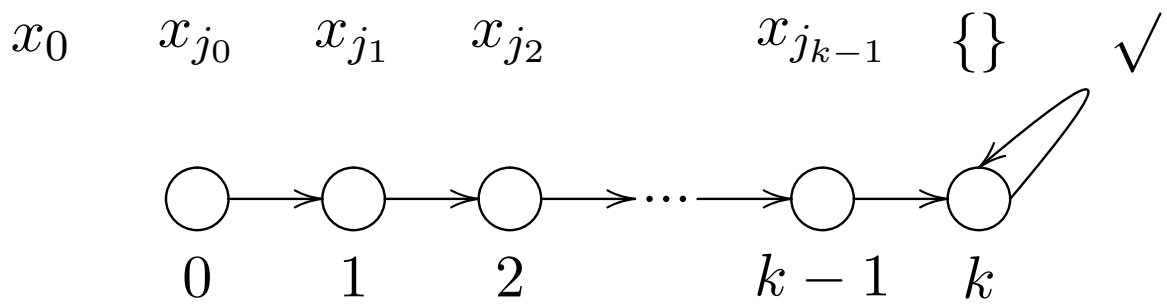
Soundness continued:

If b ends in a tick from LOOP then let $S = \{0, 1, 2, ..., k-1\}$: so it contains k states. These will correspond to $x_{j_0}, x_{j_1}, ..., x_{j_{k-1}}$.



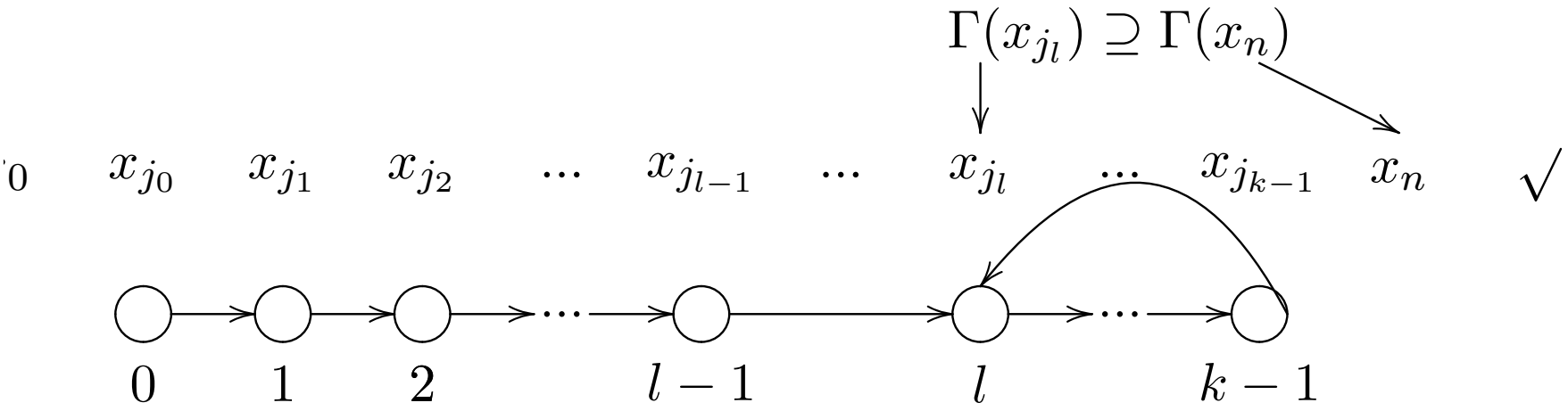
Soundness continued:

Let R contain each $(i, i+1)$ for $i < k-1$.
We will also add extra pairs to R to make a fullpath.
If b ends in a tick from EMP then just put $(k-1, k)$ and a self-loop (k, k) in R as well.



Soundness continued:

If b ends in a tick from LOOP then just put $(k-1, l)$ in R as well where l is as follows.
Say that x_m is the state that "matches" x_n. So look at the application of the LOOP rule that ended b in a tick. There is a proper ancestor x_m of x_n in the tableau with $\Gamma(x_m) \supseteq \Gamma(x_n)$ and all eventualities in $\Gamma(x_m)$ are cured between x_m and x_n.
The rule requires x_m to be just before a STEP rule. So say that $m = j_l$. Put $(k-1, l) \in R$.



Soundness continued:

Now let us define the labelling g of states by atoms in (S, R, g).
Let $g(i) = \{p \in \mathcal{L} | p \in \Gamma(x_{j_i})\}$ where, in the case of an EMP tick, $g(k) = \{\}$.

Finally our proposed model of ϕ is along the only fullpath σ of (S, R, g) that starts at 0.
That is, if b ends in a tick from EMP then
$\sigma = \langle 0, 1, 2, ..., k-1, k, k, k, k, ... \rangle$ while if b ends in a tick from LOOP then
$\sigma = \langle 0, 1, 2, ..., k-2, k-1, l, l+1, l+2, ..., k-2, k-1, l, l+1, ... \rangle$.

Change notation to deal with the two cases:

Hopefully this makes the rest of the proof easier?

Let N be the length of the first (non-repeating) part of the model: in the EMP case $N = k - 1$ and in the LOOP case $N = I$.

Let M be the length of the repeating part: in the EMP case $M = 1$ and in the LOOP case $N = k - I$.

So in either case the model has $N + M$ states $\{0, 1, ..., N + M - 1\}$ with state N coming (again) after state $N + M - 1$ etc.
In particular, $\sigma_i = i$ for $i < N$ and $\sigma_i = (i - N) mod M + N$ otherwise.

....new notation continued ....

For each $i = 0, 1, 2, ..., N + M - 1$, let $\Delta_i = \Gamma(x_{j_i})$ where we consider $\Gamma(x_{j_k}) = \{\}$ in the EMP case.

For all $i \geq N + M$, put $\Delta_i = \Delta_{i-M}$.

Thus, for $i \geq N$, $\Delta_i = \Gamma(x_{j_{(i-N) mod M+N}})$.

Helpful Lemma One

Lemma

Each Δ_i is closed under the static tableau rules (i.e. the rules except STEP, LOOP and REP).

Each Δ_i is empty or is the pre-STEP label $\Gamma(x_{j_{(i-N) mod M+N}})$.

Helpful Lemma Two

Lemma

If $X\alpha \in \Delta_i$ for some i, then $\alpha \in \Delta_{i+1}$.
Also, if $\neg X\alpha \in \Delta_i$ for some i, then $\neg\alpha \in \Delta_{i+1}$.

Just consider the $X\alpha$ case: the $\neg X\alpha$ case is similar.
Each Δ_i is empty (so not relevant here) or is the pre-STEP label $\Gamma(x_{j_{(i-N) mod M+N}})$.
After the STEP rule we will have $\alpha \in \Gamma(x_{j_{(i-N) mod M+N}+1})$.
Usually, α will stay in the Γ labels until the next pre-STEP rule at $\Delta_{i+1} = \Gamma(x_{j_{(i+1-N) mod M+N}})$.
However, ...

Helpful Lemma Two continued ...

However, when we are near the end of the branch and about to use the LOOP rule then there may be no next STEP rule.
In that case, α will stay in the labels until the end of the branch $\alpha \in \Gamma(x_n)$.
It will then also be in $\Gamma(x_{j_l}) \supseteq \Gamma(x_n)$,
So $\alpha \in \Gamma(x_{j_l}) = \Delta_l = \Delta_N$.
But as Δ_i was just before the last STEP rule before a LOOP,
$i = N + M - 1$ or some multiple of M after that.
So $(i - N) mod M = M - 1$ and $((i + 1) mod M = 0$.
Thus $\Delta_{i+1} = \Delta_N \ni \alpha$ as required.

Helpful Lemma Three

Lemma

Suppose $\alpha U \beta \in \Delta_i$.
Then there is some $d \geq i$ *such that* $\beta \in \Delta_d$ *and for all* f, *if* $i \leq f < d$ *then* $\{\alpha, \alpha U \beta, X(\alpha U \beta)\} \subseteq \Delta_f$.

As each Δ_i is closed under static rules like UNT, whenever $\alpha U \beta \in \Delta_i$ then either β will also be there or both α and $X(\alpha U \beta)$ will be.
By helpful lemma 2, if $X(\alpha U \beta) \in \Delta_i$ then $\alpha U \beta \in \Delta_{i+1}$.
Thus, by a simple induction, $\alpha U \beta$, and so also the other two formulas, will be in all Δ_f for $f \geq i$ unless $f \geq d \geq i$ with $\beta \in \Delta_d$.
It remains to show that β does appear in some Δ_d.

Helpful Lemma Three Proof continued:

If the branch ended with EMP, then we know this must happen as the Δ_f become empty.

So suppose that the branch ended with a LOOP up to tableau node x_{j_l} but that $\alpha U \beta \in \Delta_f$ for all $f \geq i$.
Sometime after i, we have $(f - N) mod M = 0$, so we know $\alpha U \beta \in \Delta_f = \Gamma(x_{j_l})$.
Thus $\alpha U \beta$ is one of the eventualities in $\Gamma(x_{j_l})$ that have to be satisfied between x_{j_l} and x_n.
Say that $\beta \in \Gamma(x_h)$ and it will also be in the next pre-STEP label x_{j_q} after x_h.
So eventually we find a $d \geq i$ such that $(d - N) mod M + N = q$ and $\beta \in \Delta_d$ as required.

Helpful Lemma Four

Lemma

Suppose $\neg(\alpha U \beta) \in \Delta_i$.
Then either 1) or 2) below hold.
1) There is some $d \geq i$ *such that* $\neg\alpha, \neg\beta \in \Delta_d$ *and for all* f, *if* $i \leq f < d$ *then* $\{\neg\beta, \neg(\alpha U \beta), X\neg(\alpha U \beta)\} \subseteq \Delta_f$.
2) For all $d \geq i$, $\{\neg\beta, \neg(\alpha U \beta), X\neg(\alpha U \beta)\} \subseteq \Delta_d$.

This is similar to the last helpful lemma.

Soundness continued:

Now we need to show that $(S, R, g), \sigma \models \phi$.
To do so we prove a stronger result. This sort of result is traditionally called a *truth lemma*.

Our lemma just says that

Lemma (truth lemma)

for all α, *for all* $i \geq 0$, *if* $\alpha \in \Delta_i$ *then* $(S, R, g), \sigma_{\geq i} \models \alpha$.

Check that this is stronger.

Proof by induction:

This is proved by induction on the construction of α.
However, we do cases for α and $\neg\alpha$ together.
For all $i \geq 0$:
if $\alpha \in \Delta_i$ then $(S, R, g), \sigma_{\geq i} \models \alpha$.
AND
if $\neg\alpha \in \Delta_i$ then $(S, R, g), \sigma_{\geq i} \models \neg\alpha$.

Proof Lemma Case p:

Fix $i \geq 0$. If $i < N$ let $i' = i$ and otherwise let $i' = (i - N) mod M + N$. Thus $\sigma_i = i'$.

If $p \in \Delta_i = \Gamma(x_{j_{i'}})$ then, by definition of g, $p \in g(i')$. So $p \in g(\sigma_i)$ and $(S, R, g), \sigma_{\geq i} \models p$ as required.

If $\neg p \in \Delta_i$ then $(S, R, g), \sigma_{\geq i} \models \neg p$ because p will not be in Δ_i (by rule X) and so we did not put p in $g(i')$.

Proof Lemma Case $\neg\neg\alpha$:

Fix $i \geq 0$.
If $\neg\neg\alpha \in \Delta_i$ then $(S, R, g), \sigma_{\geq i} \models \neg\neg\alpha$ because α will also have been put in Δ_i (by rule DNEG) and so by induction $(S, R, g), \sigma_{\geq i} \models \alpha$.

$\neg\neg\neg\alpha$ is similar.

Proof Lemma Case $\alpha \wedge \beta$:

Fix $i \geq 0$.
If $\alpha \wedge \beta \in \Delta_i$ then $(S, R, g), \sigma_{\geq i} \models \alpha \wedge \beta$ because α and β will also have been put in Δ_i (by rule CON) and so by induction $(S, R, g), \sigma_{\geq i} \models \alpha$ and $(S, R, g), \sigma_{\geq i} \models \beta$.

If $\neg(\alpha \wedge \beta) \in \Delta_i$ then $(S, R, g), \sigma_{\geq i} \models \neg(\alpha \wedge \beta)$ because by rule DIS we will have put $\neg\alpha \in \Delta_i$ or $\neg\beta \in \Delta_i$ (or one or both of them are already there) and so by induction $(S, R, g), \sigma_{\geq i} \models \neg\alpha$ or $(S, R, g), \sigma_{\geq i} \models \neg\beta$.

Proof Lemma Case $\alpha U \beta$:

If $\alpha U \beta \in \Delta_i$ then by rule UNT we will have either put both $\alpha \in \Delta_i$ and $X(\alpha U \beta) \in \Delta_i$ or we will have $\beta \in \Delta_i$.

Consider the second case. $(S, R, g), \sigma_{\geq i} \models \beta$ so $(S, R, g), \sigma_{\geq i} \models \alpha U \beta$ and we are done.

Proof Lemma Case $\alpha U \beta$ continued:

Now consider the first case: $\alpha U \beta \in \Delta_i$ as well as $\alpha \in \Delta_i$ and $X(\alpha U \beta) \in \Delta_i$.

By the helpful lemma above this keeps being true for later $i' \geq i$ until $\beta \in \Delta_{i'}$.

By induction, for each $i' \geq i$ until then, $(S, R, g), \sigma_{\geq i'} \models \alpha$.

Clearly if we get to a $l > i$ with $\beta \in \Delta_l$ then $(S, R, g), \sigma_{\geq l} \models \beta$ and $(S, R, g), \sigma_{\geq i} \models \alpha U \beta$ as required.

Proof Lemma Case $\neg(\alpha U \beta)$:

If $\neg(\alpha U \beta) \in \Delta_i$ then rule NUN means that $\neg\beta, \neg\alpha \in \Delta_i$ or $\neg\beta, X\neg(\alpha U \beta) \in \Delta_i$.

In the first case, $(S, R, g), \sigma_{\geq i} \models \neg\alpha$ and $(S, R, g), \sigma_{\geq i} \models \neg\beta$ so $(S, R, g), \sigma_{\geq i} \models \neg(\alpha U \beta)$ as required.

In the second case we can use helpful lemma four which uses an induction to show that $\neg\beta, \neg(\alpha U \beta), X\neg(\alpha U \beta)$ keep appearing in the $\Delta_{i'}$ labels forever or until $\neg\alpha$ also appears.

In either case $(S, R, g), \sigma_{\geq i} \models \neg(\alpha U \beta)$ as required.

Proof Lemma Case $X\alpha$:

If $X\alpha \in \Delta_i$ then, by helpful lemma two, $\alpha \in \Delta_{i+1}$ so by induction $(S, R, g), \sigma_{\geq i+1} \models \alpha$ and $(S, R, g), \sigma_{\geq i} \models X\alpha$ as required.

$\neg X\alpha$ is similar...
If $\neg X\alpha \in \Delta_i$ then, by helpful lemma two, $\neg\alpha \in \Delta_{i+1}$ so by induction (because we did $\neg\alpha$ first) $(S, R, g), \sigma_{\geq i+1} \models \neg\alpha$ and $(S, R, g), \sigma_{\geq i} \models \neg X\alpha$ as required.

Soundness Done:

And thus ends the soundness proof.

If we have a successful tableau then the formula is satisfiable.

One last question; where is the REP rule in the soundness proof?

Proof of Completeness:

We have to show that if a formula has a model then it has a successful tableau.

This time we will use the model to find the tableau.

Proof of Completeness:

The basic idea is to use a model (of the satisfiable formula) to show that *in any tableau* there will be a branch (i.e. a leaf) with a tick.

A weaker result is to show that there is some tableau with a leaf with a tick.

Such a weaker result may actually be ok to establish correctness and complexity of the tableau technique.
However, it raises questions about whether a "no" answer from a tableau is correct and it does not give clear guidance for the implementer.

Completeness Proof:

Suppose that ϕ is a satisfiable formula of LTL.
It will have a model. Choose one, say $(S, R, g), \sigma \models \phi$. In what follows we (use standard practice when the model is fixed and) write $\sigma_{\geq i} \models \alpha$ when we mean $(S, R, g), \sigma_{\geq i} \models \alpha$.
Also, build a tableau T for ϕ in any manner as long as the rules are followed. Let $\Gamma(x)$ be the formula set label on the node x in T.
We will show that T has a ticked leaf.
To do this we first construct a sequence $x_0, x_1, x_2, \ldots.$ of nodes, with x_0 being the root. This sequence may terminate at a tick (and then we have succeeded) or it may hypothetically go on forever (and more on that later).
In general the sequence will head downwards from a parent to a child node but occasionally it may jump back up to an ancestor.

Invariant

As we go we will also make sure that each node x_i is associated with a state $\sigma_{j(i)}$ in S.
We will ensure that for each i, for each $\alpha \in \Gamma(x_i)$, $\sigma_{\geq j(i)} \models \alpha$.
Start by putting $j(0) = 0$ when x_0 is the tableau root node.
Note that the only formula in $\Gamma(x_0)$ is ϕ and that $\sigma_{\geq 0} \models \phi$.
Good start.

Now suppose that we have identified the x sequence up until x_i.

Consider the rule that is used in T to extend a tableau branch from x_i to some children.

[EMP] If $\Gamma(x_i) = \{\}$ then we are done. T is a successful tableau as required.

[X] Consider if it is possible for the branch to stop at x_i with a cross because of a contradiction. So there is some α with α and $\neg\alpha$ in $\Gamma(x_i)$. But this can not happen as then $\sigma_{\geq j(i)} \models \alpha$ and $\sigma_{\geq j(i)} \models \neg\alpha$.

[DNEG]

So $\neg\neg\alpha$ is in $\Gamma(x_i)$ and there is one child, which we will make $x(i+1)$ and we will put $j(i+1) = j(i)$. Because $\sigma_{\geq j(i)} \models \neg\neg\alpha$ we also have $\sigma_{\geq j(i+1)} \models \alpha$. Also for every other $\beta \in \Gamma(x_{i+1}) = \Gamma(x_i) \cup \{\alpha\}$, we still have $\sigma_{\geq j(i+1)} \models \beta$. So we have the invariant holding.

[CON]

So $\alpha \wedge \beta$ is in $\Gamma(x_i)$ and there is one child, which we will make $x(i+1)$ and we will put $j(i+1) = j(i)$. Because $\sigma_{\geq j(i)} \models \alpha \wedge \beta$ we also have $\sigma_{\geq j(i+1)} \models \alpha$ and $\sigma_{\geq j(i+1)} \models \beta$. Also for every other $\gamma \in \Gamma(x_{i+1}) = \Gamma(x_i) \cup \{\alpha, \beta\}$, we still have $\sigma_{\geq j(i+1)} \models \gamma$.
So we have the invariant holding.

[DIS]

So $\neg(\alpha \wedge \beta)$ is in $\Gamma(x_i)$ and there are two children. One y is labelled $\Gamma(x_i) \cup \{\neg\alpha\}$ and the other, z, is labelled $\Gamma(x_i) \cup \{\neg\beta\}$.
We know $\sigma_{\geq j(i)} \models \neg(\alpha \wedge \beta)$. Thus, $\sigma_{\geq j(i)} \not\models \alpha \wedge \beta$ and it is not the case that both $\sigma_{\geq j(i)} \models \alpha$ and $\sigma_{\geq j(i)} \models \beta$. So either $\sigma_{\geq j(i)} \models \neg\alpha$ or $\sigma_{\geq j(i)} \models \neg\beta$.
If the former, i.e. that $\sigma_{\geq j(i)} \models \neg\alpha$ we will make $x_{i+1} = y$ and otherwise we will make $x_{i+1} = z$. In either case put $j(i+1) = j(i)$.
Let us check the invariant. Consider the first case. The other is exactly analogous.
We already know that we have $\sigma_{\geq j(i+1)} \models \neg\alpha$. Also for every other $\gamma \in \Gamma(x_{i+1}) = \Gamma(y) = \Gamma(x_i) \cup \{\neg\alpha\}$, we still have $\sigma_{\geq j(i+1)} \models \gamma$.
So we have the invariant holding.

[UNT]

So $\alpha U \beta$ is in $\Gamma(x_i)$ and there are two children. One y is labelled $\Gamma(x_i) \cup \{\beta\}$ and the other, z, is labelled $\Gamma(x_i) \cup \{\alpha, X(\alpha U \beta)\}$.
We know $\sigma_{\geq j(i)} \models \alpha U \beta$. Thus, there is some $k \geq j(i)$ such that $\sigma_{\geq k} \models \beta$ and for all l, if $0 \leq l < k$ then $\sigma_{\geq j(i)+l} \models \alpha$.
If $\sigma_{\geq j(i)} \models \beta$ then we can choose $k = j(i)$ (even if other choices as possible) and otherwise choose any such $k > j(i)$. Again there are two cases, either $k = j(i)$ or $k > j(i)$.
In the first case, when $\sigma_{\geq j(i)} \models \beta$, we put $x_{i+1} = y$ and otherwise we will make $x_{i+1} = z$. In either case put $j(i+1) = j(i)$.
Let us check the invariant. Consider the first case.
We know that we have $\sigma_{\geq j(i+1)} \models \beta$.
In the second case, we know that we have $\sigma_{\geq j(i+1)} \models \alpha$ and $\sigma_{\geq j(i+1)+1} \models \alpha U \beta$. Thus $\sigma_{\geq j(i+1)} \models X(\alpha U \beta)$.
Also, in either case, for every other $\gamma \in \Gamma(x_{i+1})$ we still have $\sigma_{\geq j(i+1)} \models \gamma$.
So we have the invariant holding.

[NUN]

So $\neg(\alpha U \beta)$ is in $\Gamma(x_i)$ and there are two children. One y is labelled $\Gamma(x_i) \cup \{\neg\alpha, \neg\beta\}$ and the other, z, is labelled $\Gamma(x_i) \cup \{\neg\beta, X\neg(\alpha U \beta)\}$.
We know $\sigma_{\geq j(i)} \models \neg(\alpha U \beta)$.
So for sure $\sigma_{\geq j(i)} \models \neg\beta$.
Furthermore, possibly $\sigma_{\geq j(i)} \models \neg\alpha$ as well, but otherwise if $\sigma_{\geq j(i)} \models \alpha$ then we can show that we can not have $\sigma_{\geq j(i)+1} \models \alpha U \beta$. Suppose for contradiction that $\sigma_{\geq j(i)} \models \alpha$ and $\sigma_{\geq j(i)+1} \models \alpha U \beta$.

[NUN] continued

We have supposed that $\sigma_{\geq j(i)} \models \alpha$ and $\sigma_{\geq j(i)+1} \models \alpha U \beta$ while also $\sigma_{\geq j(i)} \not\models \alpha U \beta$.
Then there is some $k \geq 0$ such that $\sigma_{\geq j(i)+1+k} \models \beta$ and for all l, if $0 \leq l < k$ then $\sigma_{\geq j(i)+1+l} \models \alpha$.
But then for all l, if $0 \leq l < k+1$ then $\sigma_{\geq j(i)+l} \models \alpha$.
Thus $\sigma_{\geq j(i)} \models \alpha U \beta$.
Contradiction.

[NUN] continued

So we can conclude that there are two cases when the NUN rule is used.
CASE 1: $\sigma_{\geq j(i)} \models \neg\beta$ and $\sigma_{\geq j(i)} \models \neg\alpha$.
CASE 2: $\sigma_{\geq j(i)} \models \neg\beta$ and $\sigma_{\geq j(i)+1} \models \neg(\alpha U \beta)$.
In the first case, when $\sigma_{\geq j(i)} \models \neg\beta$, we put $x_{i+1} = y$ and otherwise we will make $x_{i+1} = z$. In either case put $j(i+1) = j(i)$.
Let us check the invariant. In both cases we know that we have $\sigma_{\geq j(i+1)} \models \neg\beta$.
Now consider the first case. We also have $\sigma_{\geq j(i)} \models \neg\alpha$.
In the second case, we know that we have $\sigma_{\geq j(i)+1} \models \neg(\alpha U \beta)$.
Thus $\sigma_{\geq j(i+1)} \models X\neg(\alpha U \beta)$.
Also, in either case, for every other $\gamma \in \Gamma(x_{i+1})$ we still have $\sigma_{\geq j(i+1)} \models \gamma$.
So we have the invariant holding.

[STEP]

So $\Gamma(x_i)$ is propositionally complete and there is one child, which we will make x_{i+1} and we will put $j(i+1) = j(i) + 1$.
Consider a formula
$\gamma \in \Gamma(x_{i+1}) = \{\alpha | X\alpha \in \Gamma(x_i)\} \cup \{\neg\alpha | \neg X\alpha \in \Gamma(x_i)\}$.
CASE 1: Say that $X\gamma \in \Gamma(x_i)$. Thus, by the invariant, $\sigma_{\geq j(i)} \models X\gamma$. Hence, $\sigma_{\geq j(i)+1} \models \gamma$. But this is just $\sigma_{\geq j(i+1)} \models \gamma$ as required.
CASE 2: Say that $\gamma = \neg\delta$ and $\neg X\delta \in \Gamma(x_i)$. Thus, by the invariant, $\sigma_{\geq j(i)} \models \neg X\delta$. Hence, $\sigma_{\geq j(i)+1} \not\models \delta$. But this is just $\sigma_{\geq j(i+1)} \models \gamma$ as required.
So we have the invariant holding.

[LOOP]

If, in T, the node x_i is a leaf just getting a tick via the LOOP rule then we are done.

T is a successful tableau as required.

[REP]

Now the tricky case.
Suppose that x_i is a node which gets a cross in T via the REP rule.
So there is a sequence
$u = x_h, x_{h+1}, ..., x_{h+a} = v, x_{h+a+1}, ..., x_{h+a+b} = x_i = w$ such that $\Gamma(u) = \Gamma(v) = \Gamma(w)$ and no extra eventualities of u are satisfied between v and w that were not already satisfied between u and v.
What we do now is to choose some such u, v and w, there may be more than one triple, and proceed with the construction as if x_i was v instead of w.
That is we move on to look at the rule (as above, and the rule will not be REP) that is used to get from v to its children.
However, we use $\sigma_{\geq i}$ to make the choice of child x_{i+1} (if there is a choice).
All the reasoning above works because $\Gamma(v) = \Gamma(x_i)$ and so the invariant holds for v instead of x_i as well.
Thus we keep going.

The above construction may end finitely with us finding a ticked leaf and succeeding.
However, at least in theory, it may seem possible that the construction keeps going forever even though the tableau will be finite.
The rest of the proof is to show that this actually can not happen.
The construction can not go on forever. It must stop and the only way that we have shown that that can happen is by finding a tick.

Suppose for contradiction that the construction does go on forever.
Thus, because there are only a finite number of nodes in the tableau, we must meet the REP rule and jump back up the tableau infinitely often.
When we do apply the REP rule with triple (u, v, w) call that a jump triple.
There are only a finite number of jump triples so there must be some that cause us to jump infinitely often.
Say that (u_0, v_0, w_0) is one such.
We can choose u_0 so that for no other infinite jump triple (u_1, v_1, w_1) do we have u_1 being a proper ancestor of u_0.

As we proceed through the construction of $x_0, x_1, ..$ and see a jump every so often, eventually all the jump triples who only cause a jump a finite number of times stop causing jumps.
After that time, (u_0, v_0, w_0) will still cause a jump every so often.
Thus after that time u_0 will never appear again as the x_i that we choose and all the x_is that we choose will be descendants of u_0.
This is because we will never jump up to u_0 or above it (closer to the root).
Say that x_N is the very last x_i that is equal to u_0.

Now consider any $\alpha U \beta$ that appears in $\Gamma(u_0)$. (There must be at least one eventuality in $\Gamma(u_0)$ as it is used to apply rule REP).
A simple induction shows that $\alpha U \beta$ will appear in every $\Gamma(x_i)$ from $i = N$ up until at least when β appears in some $\Gamma(x_i)$ after that (if that ever happens).
This is because if $\alpha U \beta$ is in $\Gamma(x_i)$ and β is not there and does not get put there then $X(\alpha U \beta)$ will also be put in before the next temporal step rule.
Each temporal step rule will thus put $\alpha U \beta$ into the new label.

Now $j(i)$ just increases by 0 or 1 with each increment of i, We also know that $\sigma_{\geq j(i)} \models \alpha U \beta$ from $i = N$ onwards until (and if) β gets put in $\Gamma(x_i)$.
Since σ is a fullpath we will eventually get to some i with $\sigma_{\geq j(i)} \models \beta$.
In that case our construction makes us put β in the label.
Thus we do eventually get to some $i \geq N$ with $\beta \in \Gamma(x_i)$.
Let N_β be the first such $i \geq N$.
Note that all the nodes between u_0 and x_{N_β} in the tableau also appear as x_i for $N < i < N_\beta$ so that they all have $\alpha U \beta$ and not β in their labels $\Gamma(x_i)$.

Now let us consider if we ever jump up above x_{N_β} at any step of our construction (after N_β).
In that case there would be tableau nodes u, v and w arranged according to the jump situation.
Since u is not above u_0 and v is above x_{N_β}, we must have $\Gamma(u) = \Gamma(v)$ with $\alpha U \beta$ in them and not satisfied in between.
But w will be below x_{N_β} at the first such jump, meaning that β is satisfied between v and w.
Contradiction.

The above reasoning applies to all eventualities in $\Gamma(u_0)$.
Thus, after they are all satisfied, the construction x_i does not jump up above any of them.
When the next supposed jump involving u_0 with some v and w happens after that it is clear that all of the eventualities in $\Gamma(u_0)$ are satisfied above v.
This is a contradiction to such a jump ever happening.
Thus we can conclude that there are not an infinite number of jumps after all.
The construction must finish with a tick.
END of completeness proof.

You might like to read up on other approaches to deciding satisfiability in LTL.
Wolper [Wol85]
Schwendiman [Sch98]
Sistla and Clarke [SC85]
Schmitt and Goubault-Larrecq [SGL97]
Automata-based approaches
Resolution-based approaches.
Others, e.g. Small model theorems with axiom systems.
Implementation races, and benchmarking: [GKS10]

What about complexity?
Deciding LTL satisfiability is in PSPACE [SC85].
In fact our tableau approach can be used to show that.
Easiest to use the tableau search to directly show that the problem is in NPSPACE and then Savitch tells us also in PSPACE.
We are allowed to guess the right choices and need to show that "yes" answers can be guessed and checked using memory space bounded by a polynomial in the size of the input. To do this, just guess the right branch and remember at each step: the label here, the previous label (to check you do the STEP rule properly), the label back at an ancestor that you want to LOOP to, and the eventualities that you still have to satisfy from that. (Size of memory usage just linear in size of input even though branch length may be exponential).

And that's all for the fourth lecture.

See you tomorrow.


Valentin Goranko, Angelo Kyrilov, and Dmitry Shkatov.
Tableau tool for testing satisfiability in ltl: Implementation and experimental analysis.
Electronic Notes in Theoretical Computer Science, 262(0):113 – 125, 2010.
Proceedings of the 6th Workshop on Methods for Modalities (M4M-6 2009).


A. Sistla and E. Clarke.
Complexity of propositional linear temporal logics.
J. ACM, 32:733–749, 1985.


S. Schwendimann.
A new one-pass tableau calculus for PLTL.
In Harrie C. M. de Swart, editor, *Proceedings of International Conference, TABLEAUX 1998, Oisterwijk*, LNAI 1397, pages 277–291. Springer, 1998.


P. Schmitt and J. Goubault-Larrecq.

A tableau system for linear-time temporal logic.
In *TACAS 1997*, pages 130–144, 1997.

P. Wolper.
The tableau method for temporal logic: an overview.
Logique et Analyse, 28:110–111, June–Sept 1985.